HardDrive

USER MANUAL
FOR ALL CMD HD MODELS

FCC STATEMENT

WARNING: This equipment has been certified to comply with the limits for a Class B computing device, pursuant to subpart J of Part 15 of the Federal Communications Commissions's rules, which are designed to provide reasonable protection against radio and television interference in a residential installation. This equipment generates and uses radio and television frequency energy and if it is not properly installed, in strict accordance with the manufacturer's instructions, may cause interference with radio or television reception. If such interference is suspected, you can test this equipment by turning it off and on. If this equipment is the cause of the interference, you may correct for this by doing the following:

- Reorient the receiving antenna or AC plug of the equipment which is being interfered with.
- Change the relative positions of this device and the receiver.
- Plug this device into a different outlet, so that it and the receiver are on different circuits.

CAUTION: This equipment should only be attached to other computing devices which are equipped with shield-grounded cables and are certified to comply with Class B limits. Operation with non-certified equipment may result in interference.

This equipment should only be powered from a three-pronged grounded AC receptacle. If a multiple-outlet power box or receptacle is used to connect this device and other attached peripherals to the AC power source, the ground must be common to all units.

If necessary, contact an experienced radio-television technician for additional suggestions for connecting this equipment. You may also find the FCC booklet titled "How to Identify and Resolve Radio-TV Interference Problems" helpful. This booklet is available from the U.S. Government Printing Office, Washington, D.C. 20402, stock number 004-000-00345-4.

Copyright Notice

Copyright © 1990, 1991

by Creative Micro Designs, Inc.

4rd Edition, First printing, January, 1991

All rights reserved. No part of this document may be reproduced, in any form or by any means either manually or electronically without written permission from Creative Micro Designs, Inc.

The HD Disk Operating System (HD DOS) is protected under the Copyright Laws of the United States, and may not be copied, in whole or in part, without the written permission of Creative Micro Designs, Inc.

JiffyDOS™, RAMLink™ and HD Series are trademarks of Creative Micro Designs, Inc. Commodore 64®, 64C™, SX-64™, C-128™, C-128-D™, 1541™, 1541-C™, 1541-11™, 1571™, and 1581™ are trademarks or registered trademarks of Commodore Electronics Limited. Amiga™ is a trademark of Commodore Amiga. GEOS™, GEOS deskTop™, GEORAM™, and Berkeley Softworks™ are trademarks of Berkeley Softworks. CP/M® is a registered trademark of Digital Research Corporation. Apple® and Macintosh® are registered trademarks of Apple Computer, Inc. IBM® is a registered trademark of International Business Machines. MS-DOS® and Microsoft® are registered trademarks of Microsoft Corporation. Lt. Kernal® is a registered trademark of Xetec, Inc. and Fiscal Information, Inc

Table of Contents

Section 1: General Information

Introduction..... 1-1
Features..... 1-1
About Hard Drives..... 1-2
Accessories..... 1-2
Orders and Information..... 1-4
If You Need Technical Assistance..... 1-4
Acknowledgments..... 1-5

Section 2: Getting Started

Hookup..... 2-1
Powering Up..... 2-2
  Powerup Diagnostics..... 2-2
Powering Down..... 2-3
Low-level Formatting..... 2-3
Upgrades and HD DOS Installation..... 2-3
Partitioning & Configuration..... 2-4
Connecting Additional Drives..... 2-5
Miscellaneous Notes..... 2-5
  Setting the Clock..... 2-6

Section 3: Special Features

Default Device Numb..... 3-1
Default Partition..... 3-1
Swapping Device Numbers..... 3-1
Write Protecting the Drive..... 3-2
Date and Time Stamping..... 3-1
Front Panel Partition Selection..... 3-3
Configuration Mode..... 3-6
Install Mode..... 3-6
Auxiliary Port..... 3-7
Parallel Port..... 3-7

Section 4: Partitions and Subdirectories

Partitions..... 4-1
  Native Mode Partitions..... 4-1
  Emulation Mode Partitions..... 4-2
  Print Buffer Partitions..... 4-4
  Foreign Mode Partitions..... 4-4
Using Partitions..... 4-5
Native Mode Subdirectories..... 4-5

Section 5: Using GEOS

Getting Started. 5-1
What you need to run GEOS HD..... 5-1
Installing GEOS HD..... 5-2
Setting Up Your Hard Drive Under Geos..... 5-4
Using the HD with GEOS..... 5-5
Using QuickMove..... 5-5
Booting GEOS from the HD..... 5-7
GEOS HD Do's and Don'ts..... 5-9

Section 6: Using CP/M

Operating with CP/M..... 6-1
Installation of CP/M..... 6-1
Changing Partitions in CP/M..... 6-3

Section 7: Using Q-Link

About Q-Link..... 7-1
Signing on to Q-Link..... 7-1
Loading Q-Link from the HD..... 7-2
Q-Link Members and Backups..... 7-3
Uploading, Downloading, and Other Disks..... 7-3

Section 8: Using Commercial Software

Types of Commercial Software..... 8-1
Software without copy-protection..... 8-1
Copy-Protected Software..... 8-3

Section 9: Command Reference

Command Syntax..... 9-1
Command String Elements..... 9-1
Example command string..... 9-2
Paths in Command Strings..... 9-2
Subdirectory Paths Using JiffyDOS Commands..... 9-4
Spending Commands from BASIC..... 9-4
The Command Channel..... 9-5
Reading Disk Errors..... 9-6
Partition Numbers in File Names..... 9-8
Partition Numbers in Disk Commands..... 9-8
Partition Commands..... 9-9
Creating Partitions..... 9-9
Creating 1581 Style Sub-partitions..... 9-9
Deleting Partitions..... 9-10
Deleting 1581 Style Sub-partitions..... 9-10
Changing Partitions..... 9-10
Moving Between 1581 Style Sub-partitions..... 9-11
Formatting Partitions..... 9-11

Formatting 1581 Style Sub-partitions..... 9-12
Initializing Partitions..... 9-13
Validating Partitions..... 9-13
Partition directory..... 9-14
Renaming Partitions..... 9-14
Renaming Directory Headers..... 9-14
Getting Partition Information..... 9-15
Autobooting..... 9-16
Subdirectory Commands..... 9-17
Creating Native Mode Subdirectories..... 9-17
Moving Between Native Mode Subdirectories..... 9-18
Deleting Native Mode Subdirectories..... 9-19
Viewing Directories..... 9-20
Pattern Matching..... 9-20
Time and Date Stamped Directory Listings..... 9-21
File Commands..... 9-23
Loading Files..... 9-23
Saving Files..... 9-24
Verifying Files..... 9-25
Renaming Files and Subdirectories..... 9-26
Scratching (deleting) Files..... 9-27
Copying Files..... 9-28
Locking and Unlocking Files..... 9-30
Relative File Commands..... 9-31
Opening or Creating a Relative File..... 9-31
Positioning to a Specific Record..... 9-32
Special HD Commands..... 9-34
Software SWAP Commands..... 9-34
Software Write Protect Commands..... 9-35
Real Time Clock Commands..... 9-36
Sending SCSI Commands..... 9-39
Direct Access Commands..... 9-40
The Direct Access Channel..... 9-40
Reading and Writing Data with Direct Access..... 9-41
Block Commands..... 9-43
Allocating Blocks..... 9-43
Freeing Blocks..... 9-44
The Buffer Pointer..... 9-44
Reading Blocks..... 9-44
Writing Blocks..... 9-45
Block Execute..... 9-46
Memory Commands..... 9-46
Reading from Drive Memory..... 9-46
Writing to Drive Memory..... 9-47
Memory Execute..... 9-48
User Commands..... 9-49
U0 Utility Commands..... 9-49

Reading Blocks with U1..... 9-50
Writing Blocks with U2..... 9-50
User Jump Commands..... 9-51
Burst Commands..... 9-52
1541 / 1571 / HD NATIVE Burst Commands..... 9-52
1581 / 1581 CP/M Burst Commands..... 9-58
Burst Status Byte..... 9-64
Special Loaders..... 9-65
Job Queue Instructions..... 9-65
Job Queue Command Codes..... 9-65
Job Queue Locations..... 9-67

Appendices

Appendix A: Utilities..... A-1
Appendix B: Error Codes..... B-1
Appendix C: Partition and File Formats..... C-1
Appendix D: HD Memory Map..... D-1
Appendix E: SCSI Port..... E-1
Appendix F: Serial and Auxiliary Ports..... F-1
Appendix G: Parallel Port..... G-1
Appendix H: Power Connector..... H-1
Appendix I: Drive Specifications..... I-1
Appendix J: Command Summary..... I-1

Warranty and Return Information

Index

Problem Reporting Log

Section 1
General Information

Introduction

The CMD HD has been designed to be the easiest-to-use hard drive system available for Commodore 64 and 128 computers. Compatibility with the widest possible range of software products has been incorporated along with many special features and the ability to expand the amount of storage space used by the system. In addition, the HD may be connected to other types of computers via a SCSI (Small Computer Standard Interface) port allowing information from different computers to co-exist on the drive.

Features

The HD series of hard drives contain a number of features in addition to those considered to be necessary for the operation of a disk drive system. Here is a short list of some of these features:

- Compatibility with Commodore DOS commands, GEOS, and CP/M
- Easy connection via the serial port
- Compatibility with all serial modes and burst commands
- Real Time Clock to stamp all files with time and date
- Write protect switch on the front panel
- Swap functions allow drive to become device 8 or 9
- Up to 254 separate partitions
- 1541,1571 and 1581 emulation modes
- Native mode partition sizes up to 16 MB with true subdirectories
- Front Panel Partition Selection allows manual access to any partition
- SCSI interface port allows the connection of other SCSI devices
- Drive may be interfaced to other types of computers (Amiga, IBM, Macintosh)
- Partitions for other types of computers may co-exist on the drive

About Hard Drives

Hard drives, sometimes called "Winchester" drives or fixed disks, are used for the same purpose as are the more commonly used floppy disk drives and diskettes, this purpose being to store programs or data for use by programs. There are two main differences between hard drives and floppy drives - hard drives store much more data than do floppy drives, and the storage media (disk) is not removable from the hard drive. It is also important to note that hard drives often require special control electronics in order to operate with most systems. This is especially true with Commodore computer systems where it is also necessary to have a built-in DOS (Disk Operating System software) in the drive.

Aside from the differences between hard drives and floppy drives, differences exist between the various types of hard drives as well. The types that are most commonly referred to are RLL and MFM hard drives. These acronyms actually signify the method used for storing information on the disk. These types of drives are common on IBM-compatible computer systems where a separate controller is used by the DOS to write information to the disk. Since these drives require a separate controller, and only contain the drive mechanism and a small amount of support electronics, they are considered to be 'dumb' drives. In contrast, the SCSI (pronounced scuzzy) drive used in the HD series is considered to be a 'smart' drive because it not only contains the mechanism and support electronics, but also has an on-board processor which controls all of the specific tasks required of the drive. Although the differences between these various types of drives is not a matter of real importance to the end user, it should be noted that it is through the use of a SCSI drive and built-in controller that the HD is so easily adapted to other types of computers.

Accessories

A number of accessories are available from CMD which will extend the capabilities of your new hard drive system, or make it easier to use. The following is a list of these items, and some of the benefits of using them with your hard drive. Prices, if given, are retail and do not include shipping or tax. Prices and availability are subject to change without notice.

JiffyDOS

JiffyDOS is a disk drive speed enhancement system which will speed up access with your hard drive system. It also has a built in DOS wedge which makes sending commands to your drive much easier. JiffyDOS systems also speed up access to floppy drives. Price: $59.95 (C64), $69.95 (C128).

C-SCAN+

The C-SCAN+ is a serial bus multiplexer which allows up to 8 computers (64sand/or 128s) to access the same peripherals (hard drive, floppy drives, printers) at the same lime. This product is very useful for schools and business applications where it is desirable to have a number of computers sharing the same information files. Price: $199.95.

RAMLink

RAMLink provides a method to connect your hard drive to your computer via the hard drive's parallel connector. This speeds up disk access time considerably. RAMLink also contains a very fast RAMDOS which makes using RAM expansion units such as Berkeley Softworks' GEORAM, and Commodores 17XX series of RAM expanders much more compatible with commercial software. Additional RAM of 1 to 16 megabytes may be added to RAMLink internally. An external power supply is used allowing you to retain information stored in external RAM even when the computer is turned off. RAMLink is scheduled for release in January of 1991. Prices: Base model- $149.95.. Parallel cable- $14.95

Lt. Konnector

The Lt. Konnector is an adaptor which will allow you to attach a Xetec Lt. Kernal hard drive mechanism to the CMD HD as an add-on drive. This allows the storage capacity of the Lt. Kernal to be used under CMD's HD DOS which is much more compatible with commercial software than the DOS provided in the Lt Kernal. Price: $19.95.

Additional Drives

A full line of add-on drives in capacities of 20 to 200 megabytes are available from CMD. These are complete with case and power supply, and may be attached to your initial CMD HD hard drive system to expand its storage capacity. These drives may also be used as external SCSI hard drives with other computer systems such as the Amiga, IBM, and Macintosh. Prices: $449.95 and up.

Power Supplies

Replacement power supplies for the CMD hard drive are available from CMD. These supplies may also be used as replacements for Commodore's 1541-11 and 1581 floppy disk drives. You may also wish to have an additional power supply if you often use your hard drive at two different locations, such as at work and home. Price: $49.95.

Cables

A number of different cables are available from CMD for use with the hard drive and other peripherals. We have replacement serial cables, extended length serial cables, RAMLink parallel cables, and 25 pin cables for add-on drives. Contact CMD for prices.

Orders and Information

If you wish to place an order or need general information about any product available from us, you should call Monday through Friday 10:00 am through 5:00 pm ET. If at all possible, call before 3:00 pm ET for fastest service. The number to call is (413) 525-0023.

You may also call this number during these hours if you have a problem with an order you placed with CMD. If you have a problem with an order which you placed with one of our dealers, you should contact the dealership.

If You Need Technical Assistance

Several forms of technical assistance are available to CMD product owners. You may contact us by modem, telephone, or mail. If you experience a problem which requires immediate assistance, contact us at the number above. Whenever calling, have your serial number, model number, and any other important information ready. Jot down this information on a copy of the Problem Reporting Log found in the back of this manual.

If you have programming questions, or need non-immediate technical assistance, the best method to use is one of the two telecommunications services where support areas for CMD customers are available. One is on Q-Link, the most widely used service for Commodore computer users. Our area is located in the 'Commodore Information Network' (CIN) area of Q-Link. After entering this area, select 'Hardware Support Groups'. At the next menu, selecting 'Hardware Company Support Area' will present you with a menu containing the name of our Company, 'Creative Micro Designs'. Select this option and you will be in our area. Q-Link is online from 6:00 pm until 7:00 am ET (daily) and 6:00 pm Friday through 7:00 am Monday ET (weekends). We highly recommend this service to Commodore users.

CMD also maintains its own telecommunication service. This is available 24 hours a day at300/1200/2400 baud. The communications parameters are 8 data bits, no parity, and 1 stop bit (8-N-1 or 8-None-1). Latest updates of CMD utilities are available on this system as well as support files for BBS programs and general discussion areas. The telephone number for this service is (413) 525-0148.

Questions left to us on either of these services are usually answered within 48 hours. If you are unable to use one of these services to contact us about technical matters, contact us at our voice telephone number given earlier.

Bug reports or compatibility problems should be handled via mail as they are rarely fixable via telephone, and usually require hard copy detailing your system configuration, the software involved, and a method to repeat the problem. Use a copy of the Problem Reporting Log found in the back of this manual for reporting bugs or compatibility problems. We would also enjoy hearing about any successes you have in using your drive with various types of software. Our mailing address is:

Creative Micro Designs, Inc.
P.O. Box 646
East Longmeadow, MA 01028

Acknowledgments

The staff here at CMD are proud to release to you our series of CMD HD hard drives. Long hours were devoted to this project by many members of our staff who should not remain nameless. First and foremost, we all owe much of the credit to Mark (Codehead) Fellows, father of JiffyDOS, who was responsible for creating the DOS and overseeing much of the development; Charlie Andrews, who was responsible for developing the hardware; Doug Cotton for utilities and the manual; Charles A. Christianson who convinced us that this was a worthwhile project to start with; and Charles R. Christianson (Charlie Senior) for taking care of all the details and keeping us organized. We also thank the rest of the staff for helping take up the slack while we were so deeply involved in the design. A special note of thanks goes out to Julie, Karen, Ruth Ann, and Roxane for their support and understanding during our long hours at work.

A number of other individuals outside of CMD have been instrumental in helping us to create this system, and even though we may have been able to complete it without their help, certainly it would have taken us much longer and caused us many more headaches. For those reasons we feel that the following people have deserved honorable mention here:

Paul Bosacki for creating the GEOS patches and documentation, Miklos Garamszeghy for CP/M insight and utilities, Brian Dougherty and Matt Loveless of Berkeley Softworks for assistance with GEOS compatibility, and Quantum Computer Services and Bill Pytlovany for the Q-Link software conversion.

Section 2
Getting Started

Hookup

NOTE: Before attempting to setup and begin using your drive, you should read this manual. It may not be necessary for you to read it completely, but you should at least browse through it and become familiar with the sections which pertain to you and your intended use of the drive.

If you have not yet removed the HD from its shipping carton, do so now and place it in a well ventilated area on a level surface close to your computer and other peripherals. You should also unpack the power supply and place it in a well ventilated, out-of-the-way location. Be sure to keep the shipping carton in case you have any problems with the system and find it necessary to return your hard drive to CMD for service.

Connections

You may now connect the HD to your computer with the serial cable included in the shipping carton. This cable may be plugged into either of the connectors on the back of the HD, labeled SERIAL. The other end of this cable may be connected to the port labeled SERIAL on the back of your computer, or if this connection is already being used by another disk drive or printer, you may connect it to an available serial port on one of these devices. Refer to Figure 2-1 for the location of these connections.

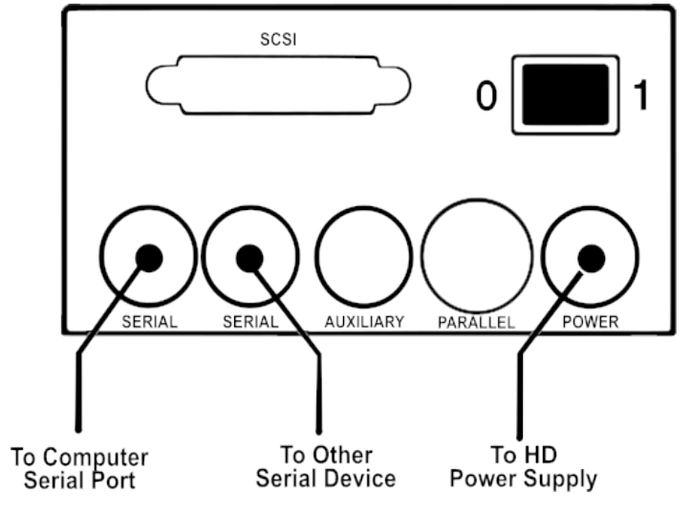


Figure 2-1

Next, connect the power supply to the HD by plugging the round DIN plug into the connector labeled POWER. Again, this is shown in Figure 2-1.

Before plugging the power supply into an AC power outlet, make sure that the power switch on the drive is in the OFF or 0 position. The power supply also has a power switch labeled ON and OFF, so be sure to place this switch in the OFF position as well. Once you have made certain that both of these switches are in the OFF position, plug the 3-prong AC power supply into a suitable grounded power outlet It is highly recommended that you plug all computer equipment (including the HD) into a power strip which contains proper surge and spike protection.

Powering Up

Your HD is now ready for use with your computer. When turning on the HD we recommend that you use the power switch located on the rear panel, leaving the power switch on the power supply in the ON position. This will also extend the life of the battery used to supply power to the Real Time Clock circuit contained within the HD.

When turning on your equipment, the order in which you activate each peripheral sometimes affects the computer's ability to recognize the existence of the devices. In this respect, the HD may be turned on first, last, or at any point in between without harm to itself. Therefore, use the order which you are accustomed to, or which seems to work best on your system.

Powerup Diagnostics

When power is applied to the HD, a test of the on-board electronics is performed to assure that the drive is in proper working condition. You should see each of the lights on the front panel activate for about one half of a second each. If a problem exists with your drive, one of the front panel indicators will stay lit and the indicator labeled ERROR will flash, unless the problem is associated with the SCSI bus or controller, in which case the ERROR indicator alone will flash. Also, only the upper ACTIVITY indicator is used in the diagnostics test, even though the lower ACTIVITY indicator may be lit during this test to indicate SCSI activity. The following list denotes which function of the drive is being tested when the each indicator is lit:

ACTIVITY	-	Serial I/O ports
ERROR	-	SCSI I/O ports
GEOS	-	Parallel I/O controller
SWAP 8	-	RAM (zero page)
SWAP 9	-	Boot ROM
WRITE PROTECT	-	RAM (all 64K)

Powering Down

When you are ready to turn off the HD, simply turn off the power switch located on the the rear panel of the drive. The drive mechanisms used in the HD are auto-parking. This means that when power is removed, the controller will automatically move the head(s) to a safe place. A utility has also been provided on the HD-UTILITIES disk to allow you manually park the heads of all drives attached to the system in case you should happen to connect any additional drives which do not auto-park. This utility, called PARK HD, may also be run to park the drive supplied in the HD if you desire to manually park the mechanism before powering down.

Low-level Formatting

Low-level formatting has already been performed on your drive mechanism before shipping from the factory, so this should not be necessary unless a problem exists. When connecting additional drives to your HD system, you should perform low-level formatting. The programs `LLFORMAT.64` and `LLFORMAT.128` have been provided for this purpose. Instructions for using these utilities can be found in Appendix A.

WARNING: Using `LLFORMAT` will destroy all data located on the drive. Never use this utility unless you are certain that it is necessary.

Upgrades and HD DOS Installation

When your drive is shipped from the factory, HD DOS is already installed. There should be no reason to re-install HD DOS on your system unless something goes wrong with the copy located in the system partition, or you receive an upgrade. In these cases, the programs `REWRITE DOS.64` and `REWRITE DOS.128` are used. These utilities will not destroy any of the data already located on the drive. Instructions for using these utilities can be found in Appendix A.

In extreme instances, it may be necessary to completely create the system from scratch. This must always be done after reformatting the drive. This step is not necessary when adding additional drives or SCSI devices to the system. If it should become necessary, the programs `CREATE SYS.64` and `CREATE SYS.128` have been provided to perform this task. Instructions for using these utilities can be found in Appendix A.

WARNING: Using `CREATE SYS` will destroy all data located on the drive. Never use this utility unless you are certain that it is necessary.

Partitioning & Configuration

The HD is shipped with a number of partitions already created on the drive. It is ready to use immediately after connecting it to your computer and power source. To take full advantage of the HD for your particular needs, it will probably be necessary for you to create some new partitions, as well as delete partitions which will not be required. You may also wish to change some of the operating characteristics of the HD. In order to facilitate these changes, the program HD-TOOLS has been provided on the HD UTILITIES disk. The use of this program is documented in Appendix A of this manual. Please refer to these instructions when using HD-TOOLS.

Configuration Parameters

The following parameters have been preset prior to shipping and may be modified by the user:

DEFAULT DEVICE NUMBER	The device number of the HD after power on or reset This has been preset to 12.
DEFAULT PARTITION	The current active partition after the HD is turned on or reset This has been preset to partition 1.

Partitions

The HD may be divided into sections called *partitions*. This is similar to having a number of different disk drives located within the same physical drive unit. As mentioned previously, the HD is shipped with a number of partitions already created for you. You may delete some or all of these if you wish, or create new partitions. Here are some general guidelines about partitions on your HD:

A. There must be at least one partition present on the hard drive in order for it to be usable.

B. You may create up to 254 partitions.

C. Partition size is dependent upon the partition type. Emulation Mode partitions contain the same number of free blocks as the drive they emulate. Native mode partitions may be from 256 blocks to 65,280 blocks in increments of 256 blocks.

At least one partition of each type has been placed on your HD prior to shipping. If you purchased a larger drive (HD-40 and up) we may have placed as many of each type as would fit on the drive.

Connecting Additional Drives

Additional SCSI drives may be attached to the HD via the SCSI port on the rear panel. This port is a DB-25S connector which has the same pin configuration as the SCSI port found on Apple Macintosh computers. Add-on drives which are configured for this type of connection are already commonly available through regular computer outlets and mail order computer stores.

When adding drives to the system, you must use the add DRIVE program (documented in Appendix A) to allow the additional drive(s) to be recognized by the HD. Up to 6 SCSI devices may be added. Also note that it is possible to purchase separate SCSI controllers which allow the attachment of more than one logical unit (or in this case drives). Using these types of 'non-embedded' controllers will allow you to connect additional drives at a much lower cost.

The SCSI host controller in the HD has been assigned as SCSI device number 7. This number may be changed later via software if you need to add another SCSI device which must be number 7. The drive in the HD has been set as SCSI device number 0. The only way to change this number is by opening your unit and changing a jumper on the drive, which will void your warranty. This should not be a problem, as most any SCSI device may be jumpered to use a different device number.

When shopping for add-on SCSI devices, try to find out if the device number may be changed with a jumper arrangement. This will help you avoid any device number conflicts. The HD's total system capacity may be up to 4 Gigabytes. Note that additional drives or other SCSI devices which you add to the system must have their own source of power.

Miscellaneous Notes

The CMD HD series of hard drives are very complex computing devices, but every opportunity has been taken to design them to withstand normal use in standard computing environments.

The Real Time Clock Battery

The CMD HD has a built-in battery-backed real time clock circuit which allows for time and date stamping of all files. The battery which powers this circuit when the HD is not turned on has a life expectancy of approximately ten years. This battery is not user replaceable, and if it should fail after the warranty has expired, it should be replaced only by a qualified electronics technician.

Setting the Clock

A utility program for setting the clock, `SET HD CLOCK`, is located on the floppy disk labeled HD UTILITIES which was included with your drive. This program will present you with the current clock setting when it begins. The clock has been set prior to shipping the drive, but since you may live in a different time zone, it may be necessary for you to change this setting. Also, note that this program is in BASIC, and may serve as an example of how to read and set the clock from within your own programs.

Other Operating Characteristics

While operating your drive, you may note some of the following characteristics. These characteristics arc normal, and should not cause you to worry or be alarmed.

MOTOR NOISE — Unlike floppy disk drive units, the motor in a hard drive spins constantly. This will cause a constant whirring sound which is louder in some drives than in others.

HOUSEKEEPING — Some of the hard drive mechanisms employed within the HD perform periodic housekeeping routines. Part of this activity is to exercise the drive's head mechanism, which causes a chattering type of sound. This will occur once every five to ten minutes when the drive is not active.

MOTOR HEAT — Due to the fact that the motor is constantly spinning, and since the drive mechanism is mounted near the top of the case, you may notice that the top of the drive case gets warm or even hot under constant use. This should not be harmful to the drive as long as you have allowed for proper ventilation by leaving an inch or two of space all around the drive. Our drive enclosure has been examined and approved by the engineering departments of the companies which supply the drive mechanisms.

Section 3
Special Features

Default Device Number

The device number of your CMD HD hard drive is preset to 12 at the factory. Unlike other disk drives used with Commodore computers, this device number is not controlled by hardware, but is instead kept in a table located in the system partition on the HD. You may reassign the HD's default device number to any device number from 8 to 29 by using the HD-TOOLS utility located on the HD UTILITIES disk. If for any reason the operating system will not load properly, the drive may fall back into its boot configuration, in which case it will assign itself device number 30.

Default Partition

The default partition is the partition in which the HD places itself after RESET has been pressed or power has been applied and is preset at the factory to partition number 1. You may change the default partition at any time by using the HD-TOOLS utility. When changing this default, make sure that you choose a partition which is actually in existence on the HD.

Swapping Device Numbers

One of the most powerful and unique features of the HD is its ability to swap device numbers with other disk drives on the serial bus. This function is implemented through the use of the SWAP 8 and SWAP 9 switches on the front panel. Pressing SWAP 8 or SWAP 9 immediately makes the HD device 8 or 9 and assigns the HD's default device number to the drive that is normally device 8 or 9. The red indicator above the appropriate SWAP button will light when the HD has been swapped to device 8 or 9.

As an example, assume your HD is device 12 and there is a 1541 drive on the serial bus that is device 8. When SWAP 8 is pressed, the HD becomes device 8 and the 1541 is swapped to device 12. Pressing SWAP 8 again restores the HD and 1541 to their original device number assignments.

The most common use for swapping device numbers is when using software that will recognize only devices 8 and/or 9. With such programs, the SWAP function enables you to make the HD device 8 or 9 before loading the software (if the program is on the HD), or after loading (if the program is copy-protected and cannot be moved onto the HD). In either case, having the ability to instantly make the HD device 8 or 9 guarantees that any data files used by the program can be stored on the HD.

Things to remember about device swapping

1. When first powering on or resetting a C-64 (or entering C-64 mode on a C-128), the serial bus must be cleared before the SWAP functions can be used. This can be done by performing any disk access (such as LOAD"$",8) or by the command POKE 56576,199.

2. When GEOS is active, the SWAP functions cannot be used. This is because GEOS is always controlling an active handshake with the current drive on the serial bus. In any case, swapping device numbers after GEOS has booted would not be recommended because GEOS assigns a different device driver to each type of drive. Therefore, the proper time to swap device numbers when using GEOS is *before* booting.

3. NEVER press the SWAP buttons while a drive on the serial bus is being accessed. Doing so invites the possibility of locking up the bus, making it necessary to power down or reset to clear the condition. This could jeopardize the contents of any files being written at the time. A role-of-thumb to apply before pressing the SWAP buttons is to make sure that none of your drive activity lights are on, and that the program you are using is not processing data at the time. (A good time to swap device numbers is while a program is waiting for user input).

Write Protecting the Drive

The write protect function provides you with a way to prevent any unintentional writes to the HD. To activate the write protect function, press the WRITE PROTECT button on the front panel. The red indicator light above the WRITE PROTECT button will light when the HD is write protected. To deactivate this function, press WRITE PROTECT again.

The write protect function can be useful when testing software on the HD for the first time or when you are formatting a disk in another drive on your system and you want to make absolutely sure that the formatting will not accidentally take place on the HD.

Date and Time Stamping

The HD circuitry contains a battery-backed real time clock circuit, which allows the operating system to time and date stamp each file entry within the directory. This is a feature which has long been desired by users of the Commodore computers. In order for you to take full advantage of this feature, the HD has been equipped with several new commands which enable you to view the time and date of Hies, and to use the clock from your own programs. These commands are described later in the Command Reference Section of the manual.

Front Panel Partition Selection

You may occasionally run into situations where it is necessary to change from one partition to another, but are unable to do so from within the current program. This usually arises when using a program which does not allow you to send disk commands, or only allows you to send specific disk commands from a menu. Many multiple-disk programs require you to change disks at some point, and then attempt to verify that this has been done by reading the disk name or the disk ID.

To help get around these types of problems, the switches on the front panel of the HD may be placed into a special mode, allowing you to change the current partition. This feature has been created without adding any new hardware to the HD, and while it may not be as easy to use as most of the HD's other features, it is flexible enough to perform the required task without increasing the overall cost of the product.

Using FPPS Mode

FPPS (Front Panel Partition Selection) Mode makes special use of three of the switches which are located on the front panel of the HD - SWAP 8, SWAP 9, and WRITE PROTECT. Here is a quick overview of what each switch is used for in this mode.

WRITE PROTECT	This switch has two functions in FPPS Mode. It is used to enter and exit the mode, and is also used to set the lowest digit (ones) in the partition number you are changing to.
SWAP 9	This switch is used to set the middle digit (tens) in the partition number you are changing to.
SWAP 8	This switch is used to set the highest digit (hundreds) in the partition number you are changing to.

When setting digits with these switches, quickly press and release the switch. No change will occur until the switch is released, and holding the switch down for too long of a period of time may cause undesired effects. The indicator lamps for GEOS, SWAP 8, SWAP 9, and WRITE PROTECT are used to indicate the selections you make while in this mode.

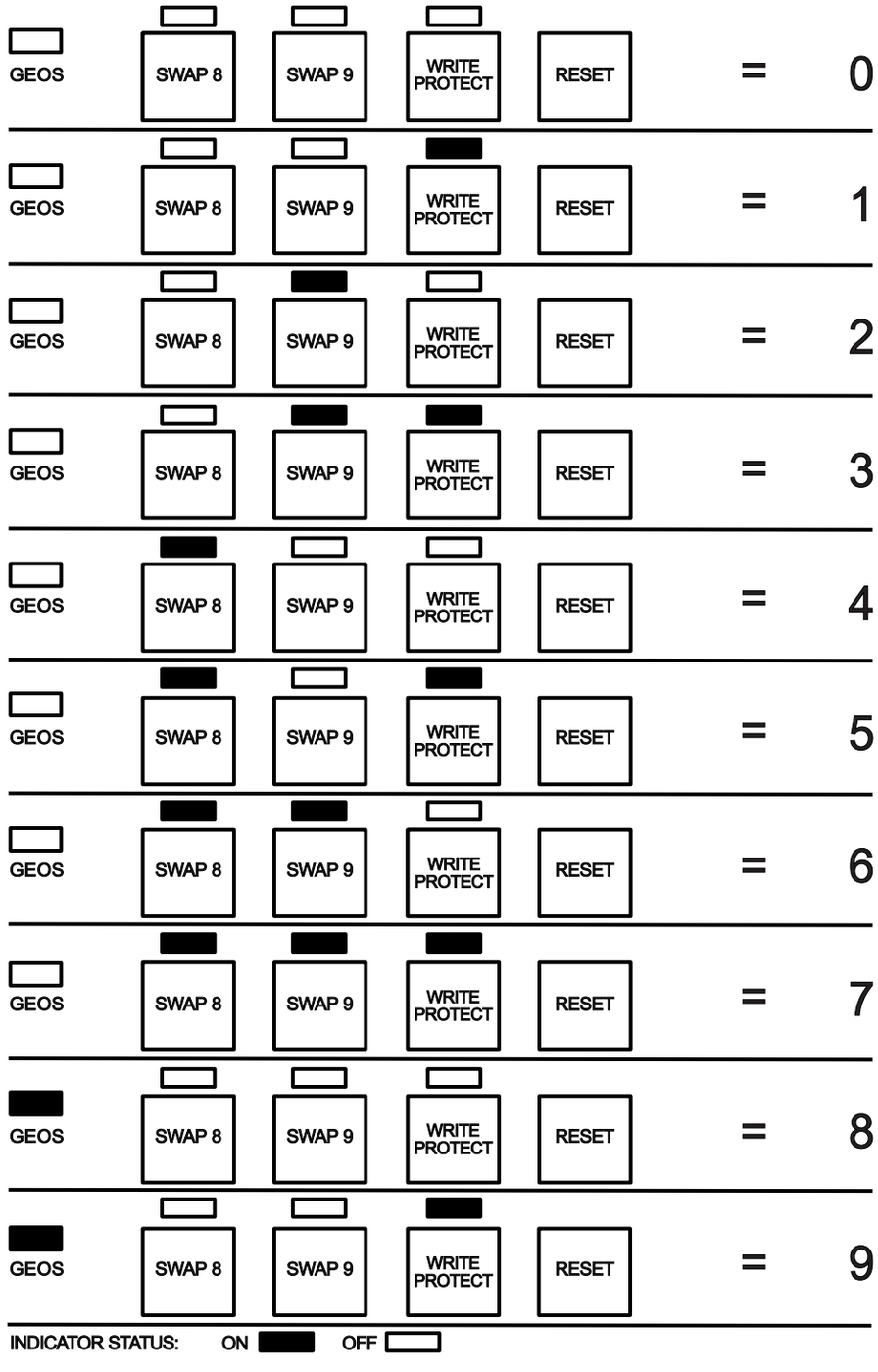
Selecting a Partition with FPPS

The following example will show you how to use FPPS Mode to change partitions. Try this example yourself, then practice using the mode with different partitions on your HD.

1. Enter FPPS Mode by holding down the WRITE PROTECT switch until the upper activity light comes on. This takes about 2 seconds. When the upper activity lamp glows green, release the WRITE PROTECT switch. The lamp should now begin flashing to indicate that you have entered FPPS Mode successfully.
2. Quickly press and release the WRITE PROTECT switch once. The indicator lamps for GEOS, SWAP 8, SWAP 9, and WRITE PROTECT should all be off.
3. Quickly press and release the WRITE PROTECT switch once again. This time you should see the indicator for WRITE PROTECT come on. This means that the 'ones1 digit of your three-digit partition number is now set to 1. The numbering scheme used for the indicator lamps may be seen in the chart on the following page.
4. Continue to quickly press and release the WRITE PROTECT switch a number of times, each time checking the indicator lamp status against the chart. Move on to the next step when this digit value is set to 1 again (only the WRITE PROTECT lamp is lit).

   Since you may have up to 254 partitions on the HD, 3 digits are required to specify a partition number. You have already set the ones digit to a value of 1, specifying partition number 001. If you were to exit FPPS Mode at this time, the HD would change to partition number 1.
5. Quickly press and release the SWAP 9 button. Notice that the WRITE PROTECT lamp turned off? As soon as you press and release a button for the first time (or after you press one of the other buttons), it will show the current status of the digit controlled by that particular switch. Since all digits are cleared upon entry into the mode, the first press and release of each switch will show a 0 for that digit Continue to step through the numbers for the tens digit using SWAP 9 until this digit equals zero (all lamps off), then move on to the next step.
6. Repeat the last step, this time using SWAP 8 instead of SWAP 9. Notice that you can only set this digit as high as 2 before it becomes 0 again? This is because partition numbers above 254 are not legal. The tens digit will be limited as well if you were to set the hundreds digit to 2. The same goes for the ones digit if the hundreds digit is set to 2 and the tens digit is set to 5. Play around with these combinations if you wish, but when you are done, make sure you are set for partition number 001 by pressing each of the FPPS Mode switches one by one to verify the status.
7. Exit FPPS by holding down the WRITE PROTECT switch until the upper activity lamp turns off. Then release the WRITE PROTECT switch. You should now be located in partition number 1.

FRONT PANEL PARTITION SELECTION MODE INDICATOR CHART



Configuration Mode

Whenever you wish to make changes to your system defaults, create or delete partitions, or place a new version of the DOS on your drive, you will need to place your drive into the Configuration Mode. This mode is also used to rewrite the current version of HD DOS in the event that anything should go wrong with the existing copy. To enter Configuration Mode, press and hold down the WRITE PROTECT switch on the front panel of the HD. While continuing to depress this switch, press and release the RESET switch. This action will cause the ACTIVITY indicators to flash for a second or two. After this activity has stopped, you may release the WRITE PROTECT switch. The drive is now ready to be configured. You may load and run the required software from the HD UTILITIES disk. HD-TOOLS is used to create or delete partitions and to change system defaults. REWRITE DOS is used, as the name implies, to rewrite the DOS to your drive. Both of these utilities are found on HD UTILITIES, the floppy disk that was included with your system. After you have completed making changes, you may place the drive back into it's normal mode of operation by pressing RESET.

Install Mode

If for any reason the system parameters become corrupted and it is no longer possible for the system to boot, it may be necessary for you to re-install a complete operating system on your drive. Before doing this you should always check to see if it is possible to correct the problem by re-writing the DOS onto the drive (see the preceding paragraph describing Configuration Mode). If re-writing the DOS fails to solve the problem, the next step is to place the drive into Install Mode and run the CREATE SYS utility located on the HD UTILITIES disk.

WARNING: USE THIS PROCEDURE ONLY AS A LAST RESORT. THIS PROCESS WILL DESTROY ALL DATA ON YOUR DRIVE!

Begin this process by turning on your computer and drive. Press and hold down both the SWAP 8 and SWAP9 switches located on the front panel of the drive. While continuing to hold down both SWAP switches, press and release the RESET switch on the front panel of the drive. This action will cause the ACTIVITY indicators to flash fora second or two. After the drive activity has stopped, you may release the SWAP switches. The drive is now ready for re-installation of the system. To do this, load and run the CREATE SYS utility from your floppy disk drive. After loading some required data, the program will create a new system partition on your drive and install the version of HD DOS located on the HD UTILITIES disk. This process should only take a couple of minutes to complete. When the program has finished and a READY prompt appears on your screen, press the RESET switch to initialize the drive.

Before you can begin using the system again, you will need to use HD-TOOLS to create partitions on the drive. Also, you will need to use HD-TOOLS if you wish to change the default device number and default partition values (CREATE SYS sets these defaults to 12 and 1 respectively). If for some reason you are still having problems with the drive, this may be due to a hardware failure and you should contact CMD for further instructions.

Auxiliary Port

The Auxiliary Port on the rear of the drive is not being used by the current version of the operating system. Future revisions of the DOS will use this port as a dedicated printer port. When this function is implemented, the HD will intercept and store data normally sent to devices 4 or 5 in a special partition, and then spool it out to the printer connected to the Auxiliary Port Since the HD is able to receive and store data much faster than it can be printed, this feature will greatly enhance the speed of all print operations, and quickly free up your computer so that it can be used for other tasks.

Parallel Port

The Parallel Port on the rear of the drive is used to connect the HD to a similar parallel port on CMD's RAMLink which plugs into the C-64/128. HD users desiring even faster performance with the HD can easily boost data transfer speed to the maximum attainable level while retaining compatibility.

Section 4
Partitions and Subdirectories

Partitions

The entire storage area of a CMD HD hard drive system may be divided into a number of smaller areas called partitions. While this term is very common to users of larger computer systems, it may be new to many users of Commodore computers. Simply stated, the use of partitions on the HD gives the appearance of using a number of disks, all located within the same drive. The figure below helps to illustrate how the HD's total storage area is split up into separate disks. In all, the HD can handle up to 254 of these partitions. As mentioned earlier, your HD was shipped with a number of these partitions already installed. Partitions are created and deleted by using the HD-TOOLS utility. Each partition may be defined to be one of several types. The four types currently available for use on the HD are:

Native Mode
Emulation Mode
Print Buffer
Foreign Mode

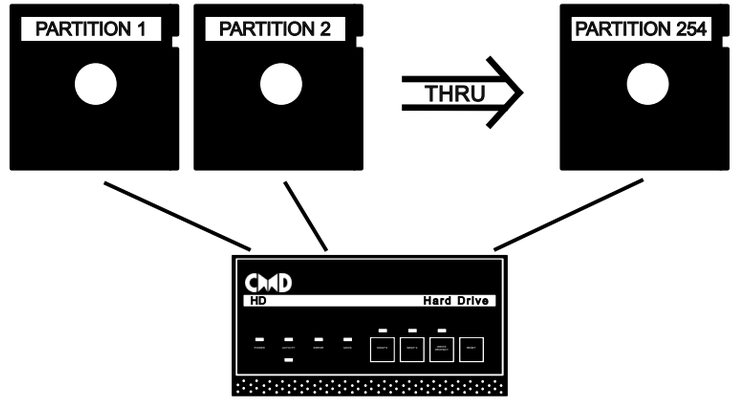


Figure 4-1

Native Mode Partitions

Native Mode partitions allow you to take full advantage of the many additional features provided by CMD's HD DOS (Hard Drive Disk Operating System) while retaining full compatibility with all standard Commodore DOS commands. This type of partition can access up to 16 Megabytes of storage space, providing the highest capacity available to Commodore DOS

commands on the HD. Since Native Mode partition size is variable and may also be as small as 256 blocks, it allows you to use only the amount of storage which you feel is required for a particular partition. This mode is also the only mode which supports true subdirectories and dynamic allocation of directory space. This means you can easily organize your files and continue to add files until no free blocks remain in the partition.

Native mode partitions have 256 sectors per track, and may have from 1 to 255 total tracks. Since the size is variable, all header, BAM, and directory information must be stored on track 1. Subdirectories have a file type of DIR and have been assigned a flletype value of 6.

Emulation Mode Partitions

Emulation Mode partitions allow you to retain compatibility with software programs which require the tracks and sectors of a disk, as well as the BAM and directory, to be laid out in the same way as on a particular type of Commodore floppy disk drive. For this reason, an Emulation Mode partitions has a fixed storage capacity equal to the capacity of the disk drive that it is emulating. There are four types of Emulation Mode partitions available on the HD. They are:

1541 Emulation Mode
1571 Emulation Mode
1581 Emulation Mode
1581 CP/M Emulation Mode

As the names imply, these partitions emulate Commodore's popular 1541, 1571 and 1581 disk drives. This is accomplished by utilizing the same track and sector layout as the type of disk drive being emulated. The BAM and directory areas of these partitions are also located in the same blocks as they would be on the emulated drive. Even the internal job queue codes and locations have been duplicated. Other similarities and a few beneficial differences have been created within the Emulation Mode partitions. It is important to note that emulating these other disk drives fully in the areas of hardware and firmware mapping would have driven the cost of the HD to an unreasonable level, and was therefore not attempted. All other aspects of compatibility were carefully scrutinized, and incorporated where feasible. The following paragraphs describe each of the individual Emulation Modes.

1541 Emulation Mode Partitions

In 1541 Emulation Mode partitions the directory and BAM are found in the same locations as they are on the Commodore 1541. All bytes within these blocks have been defined identically to their counterparts on the 1541, including the BAM bytes. This type of partition uses 684 blocks of disk space (342 SCSI blocks) of which 664 are free for user data or programs.

There are some differences between the standard 1541 and the HD's 1541 emulation. First, it is fully possible to use fast serial with this emulation mode, and this is in fact done automatically when communicating with fast serial compatible equipment Another difference is that 1541 Emulation Mode allows the use of 1571 burst commands. These two features allow 1541 partitions to be utilized as single sided 1571 disks.

The way in which relative (REL) files are handled can at times be different as well. In the standard 1541, only six relative file side sectors may be used. This is also the default method of handling REL files in 1541 Emulation Mode partitions, but it is also possible to read and write the 'super' side sector REL file format used on the 1581 disk drive. Special programming is be required to create such a file in 1541 partitions, but once it is created it is handled properly without further intervention.

One further difference exists, and this concerns time and date stamping of all flies. This is accomplished in the directory file entry of each file in the same manner employed by GEOS. By using this method, all files created on the HD (including those created under GEOS), will report the proper time and date of creation when listing the directory with the special directory commands. Any time a file is appended to, copied, or renamed, a new time and date stamp will be placed on the file.

1571 Emulation Mode Partitions

1571 Emulation Mode partitions are identical to 1541 Emulation Mode partitions with only a few differences. First, 1571 partitions have twice as much storage capacity as do the 1541 partitions. This type of partition uses 1366 blocks of disk space (683 SCSI blocks) of which 1328 are free for user data or programs.

There are also a number of extra bytes required for the BAM in the header block and on track 53. These have been allocated in the same fashion as they are on the standard 1571 disk drive. Also, 1571 partitions are always equivalent to the double-sided version of the 1571. If for some reason you need to emulate a single-sided 1571 disk, the 1541 partition is fully capable of doing this and should be used instead.

As with 1541 Emulation Mode partitions, it is possible to read and write to the 'super' side sector form of REL files. This method, if used, allows REL

files created within 1571 Emulation Mode partitions to grow beyond the former maximum size of 726 blocks.

When using CP/M on 1571 partitions, only the Commodore CP/M format is supported. Programs which attempt to reprogram the disk controller chip on the standard 1571 in order to read and write in MFM formats will not work with 1571 Emulation Mode partitions since this controller is not present in the HD.

1581 and 1581 CP/M Emulation Mode Partitions

1581 Emulation Mode partitions provide excellent emulation of the 1581. All header, BAM, and directory information is identical to the 1581. Fast serial and most burst commands are fully compatible. The main area of difference between the 1581 and its HD counterpart is that the HD does not utilize the track cache buffer found on the standard 1581. This was incorporated into the 1581 to increase speed- an area where the HD does not suffer. All other standard DOS commands are fully implemented, including 1581 style partitioning commands. In order to retain full compatibility with the 1581, the DOS initialize command initiates a change in the current 1581 partition status, causing further disk accesses to be performed in the root directory of the 1581 partition.

If you wish to use a 1581 partition to boot CP/M, you must use the 1581 CP/M Emulation Mode. This is because the boot block of the 1581 (track 1, sector 0) is internally translated to a different location on the disk (track 40, sector 5) under CP/M. We're not entirely certain why Commodore chose to do this, but in order to eliminate possible conflicts, we decided to create a special partition type. It is unwise to use this type of partition for storage of standard files or GEOS files, as this could eventually lead to corrupting the directory. See 'Operating with CP/M' in section 3 for more information concerning this.

Print Buffer Partitions

Print Buffer partitions are not currently usable by the system. This feature will be available in a later version of HD DOS. hd-TOOLS will allow you to create these partitions, but none of the print spooling features have yet been implemented in the DOS itself.

Foreign Mode Partitions

Foreign Mode partitions are used to define an area of the drive which is to be used for data from another type of computer. Foreign Mode partitions are currently usable, and the partition info utility has been provided to allow you find out where partitions on the HD are located. With this information, it should be possible for you define the accessible areas of the HD on other types of computers. To do this, you must use partitioning software intended for that type of computer.

Using Partitions

The CMD HD hard drive can be sectioned off or divided into a number of smaller areas called partitions. In all, the HD can handle up to 254 of these partitions. As mentioned earlier, your HD was shipped with a number of these partitions already installed. Partitions are created and deleted by using the `HD-TOOLS` utility.

Current Partition

Partition 0 (zero) has a special meaning under HD DOS. It is used to indicate the current partition (the partition in which the HD is currently working). This allows for compatibility with software which issues a "0:" within filenames or disk commands. Before attempting to use most commercial software with the HD, it is usually wise to place the drive into the partition you wish to use. This will help to ensure that file accesses will occur within that partition, especially if the software does not give you the option of sending disk commands.

Native Mode Subdirectories

The following information is intended as an introduction to how Native Mode subdirectories are stored on the HD. The commands used to create, remove, and move around within subdirectories can be found in the Command Reference Section of this manual.

Native Mode subdirectories are similar in structure to the subdirectories used on MS-DOS types of computers. When a subdirectory is created, a DIR type file (filetype 6) is created and added to the current directory. Subdirectory names may be up to 16 characters long, just as any other filename. This "file" is initially two blocks long, and consists of a directory header block and the first directory block. These blocks are always located next to each other on the same track, and if two adjacent blocks cannot be found, no directory will be created

The storage space available to subdirectories is the same as that available to the parent directory (the directory in which the subdirectory exists). In fact, all of the blocks within a Native Mode partition are shared between all directories within that partition. This is quite different than the method used for 1581 type subdirectories (or sub-partitions as they are referred to in this manual). This means that if there are 62000 blocks free in the partition, this number of blocks free will be indicated no matter which directory you are located in. If a 37 block file is saved in any directory within the partition, all directories within that partition will indicate 37 fewer blocks.

Subdirectories may be created in the 'root' directory (the first or main directory in that partition) or within another subdirectory. Placing a

subdirectory within another subdirectory is called 'nesting'. There is no actual limit to the number of directories located in a partition, nor is there any limit to how deep subdirectories may be nested. The only limitation on creating subdirectories is the number of adjacent blocks located within the partition. There are, however, some practical limitations if you wish to be able to easily access files located in various subdirectories with a single command. This is because the input buffer of the HD is only 254 characters long, therefore nesting subdirectories too deep could necessitate using more than one command string to access files within a particular directory.

Section 5
Using GEOS

Getting Started

This chapter will take you step by step through the procedures for getting your new CMD HD up and running under all 2.0 versions of GEOS. This includes GEOS 128, and the GEOS supplied with Berkeley's new GEORAM. Unless differences exist, all versions of GEOS will be referred to as GEOS.

In this chapter, you will learn how to:

- Replace the current CONFIGURE on your boot disk with the appropriate version from the GEOS Utilities disk supplied with your new CMD HD.
- Move HDTime to your boot disk.
- Use QuickMove to quickly copy files across partitions, as well as move into another partition.

This chapter also offers some advice on effectively using the HD under GEOS. Any problems that you might encounter, and a short list of Do's and Don't's appear at the end of this section.

What you need to run GEOS HD

Required Software:

- Any 2.0 version of Berkeley Softworks' GEOS (boot disk)
- GEOS Utilities from CMD

Required Hardware:

- Commodore64 or 128
- 1541 or 1571 disk drive
- CMD HD
- Mouse or joystick

Optional Equipment:

- A Ram Expansion Unit, either Commodore's or Berkeley's. An REU dramatically improve the performance of GEOS, and easily facilitates the use of multiple drives of different types (ie. 1541 and 1581).
- Additional drives. For example, Commodore's 1581 3.5" disk drive.
- A GEOS supported printer. A list of supported printers can be found in your GEOS 2.0 manual.

Installing GEOS HD

Installing GEOS HD is not a complicated procedure. However, because you will be replacing files on your boot disk, we suggest you read this section over until you are comfortable with the procedures.

The following steps summarize the procedure necessary to install the CMD HD under GEOS:

1. First, with your CMD HD disconnected, power up your 64 or 128 and boot GEOS as you would normally.
2. Remove the GEOS System disk from Drive A and insert the GEOS Utilities disk supplied with your CMD HD. Click on the disk drive icon representing Drive A. This will open GEOS Utilities on the DeskTop.
3. From the menu bar, select disk. After the disk menu unfolds, click on `validate`. At this time, the DeskTop will validate GEOS Utilities. If the DeskTop reports an error, proceed no further with the installation procedures! Contact CMD for a replacement disk.
4. Assuming the Desktop reported no errors, remove GEOS Utilities from Drive A, and insert your GEOS boot disk. Click on the drive icon to open the disk.

At this point, it is necessary to remove the CONFIGURE program that came originally with GEOS and replace it with the appropriate version found on GEOS Utilities. If you do not possess a backup of this file, back it up now! Please refer to your GEOS 2.0 manual for instructions on how to back up files.

5. Place the pointer over the CONFIGURE icon and click once. At this point the icon will become highlighted. After a short pause, click again. A copy of the icon, a 'ghost icon1, will appear attached to the pointer.
6. Move the pointer to the bottom border area of the screen, between the waste basket and the printer icon. Click once. Now, the icon will disappear from the notepad, and reappear on the border.
7. Again, place the pointer over the CONFIGURE icon and click once. As before, the icon will become highlighted; click again and move the ghosted icon over to the waste basket.
8. With the CONFIGURE icon positioned over the waste basket, click once. Your drive will whirr for a few moments, and then the CONFIGURE icon will disappear from the border. The name of the file now appears under the waste basket indicating that the file has been deleted.

9. Remove the GEOS System disk from Drive A and insert GEOS Utilities. Open it by clicking on the disk icon.

   Across the top of the notepad, you will find four CONFIGURE files. The CONFIGURE file you select will be determined by two factors:

   1. Whether you are running GEOS 128 or GEOS for the 64.
   2. Whether you own an REU, and if you do, whether it is a Commodore REU or Berkeley's geoRAM.

   If you are running GEOS for the 64:

   a. Select CONFIGURE if you do not have an REU.
   b. If you have a Commodore REU (1764 or 1750), also select CONFIGURE.
   c. If you own a geoRam, select CONFIGURE_r.

   If you are running GEOS 128:

   a. Select 128 CONFIGURE if you do not have an REU.
   b. If you have a Commodore REU (1764 or 1750), also choose 128 CONFIGURE.
   c. If you own a geoRam, select 128 CONFIGURE_r.

10. Once you know which CONFIGURE file is appropriate to your version of GEOS, place the pointer on the appropriate CONFIGURE and click once. Click again and drag the ghosted icon to the border. Click to deposit the CONFIGURE icon on the border.
11. Remove the GEOS Utilities disk, and insert your boot disk. Click on the Drive A icon to open it
12. Place the pointer atop the CONFIGURE icon and click once. Drag the ghosted icon to the notepad and click again.
13. The DeskTop will now request that you insert GEOS Utilities in Drive A. Do so, and click on OKAY. After a few moments, you will be requested to insert your boot disk. You will have to swap disks a number of times before the copy operation is completed

13. Once the procedure has been completed, the new CONFIGURE will appear on your boot disk. IMPORTANT: If you copied over one of the CONFIGURE.r files, you must rename the file `CONFIGURE` or `128 CONFIGURE` before using the boot disk.

In order to ensure the proper operation of your HD under GEOS, there is another file that must be placed on your boot disk. This file is HDTime. HDTime is an autoexec file. On booting GEOS, it ensures that the HD is in a 1581 partition. In addition, it sets the GEOS system clock with the time

and date found in the HD's real time clock chip. The same version of HDTime runs under all versions of GEOS 2.0.

To install HDTime on your boot disk, follow the procedure described above for moving CONFIGURE to your boot disk. If there is not enough room on your boot disk to accomplish this, we suggest that you delete 3 or 4 printer drivers that you are not likely to use. To delete these, either follow the procedure described above for deleting the original CONFIGURE file, or refer to your GEOS 2.0 manual.

At this time, shut down your system and reconnect you CMD HD.

Setting up Your Hard Drive Under GEOS

With your new CONFIGURE and HDTime copied to your boot disk, you are ready to set up your HD for use under GEOS. Before you reboot, it is important to note the following limitations:

- Currently, only 1581 Emulation Mode partitions are supported.
- CONFIGURED 1581 DirShadow option is no longer supported by the 1581 driver.
- GEOS will only recognize drives numbered 8, 9 or 10. If your CMD HD is at some other device number, GEOS will not be able to find it. Either press the SWAP 9 switch on the front panel of the CMD HD, or reconfigure your HD to another device number.
- GEOS expects device numbers to be consecutive. In other words, on a two drive system without ram expansion, the HD must be device 9. Under these circumstances, GEOS will not find an HD at device 10.
- Because GEOS expects to be booted from a floppy at device 8, the CMD HD cannot occupy device 8.

With these limitations in mind, reboot GEOS with the HD powered up. It is not likely that when the DeskTop comes up the first time that you will see an additional drive icon associated with the HD. To accomplish this:

1. Double click on the CONFIGURE icon. After a short pause, the CONFIGURE application screen will open.
2. Assuming the device number of the HD is 9, select a 1581 as Drive B. The activity light on the HD will briefly light.
3. Select `save configuration` from the file menu to install the HD under GEOS.
4. Your HD is now installed and ready for use under GEOS. Quit from CONFIGURE, and explore the possibilities offered you through your new CMD HD.

Using the HD with GEOS

You will find using the CMD HD under GEOS transparent. In fact, your CMD HD will behave exactly as a 1581 with one notable exception- speed. Simply put, your CMD HD is the fastest disk device ever to run under GEOS. Applications and data files will load at impressive speeds. An REU and CMD HD remove the last vestige of complaint from the GEOS operating system. No one using an REU and HD will ever again be able to claim that GEOS is too 'slow'.

With QuickMove, the CMD HD offers you access to multiple 1581 partitions. Individual partitions can be dedicated to a single application and support files. For example, one partition may alone be dedicated to geoPublish with associated applications and clip art Another might contain a complete applications development environment. Yet another to word processing or graphics. And with QuickMove, copying files between partitions is made effortless - the way it should be!

For that reason alone, a copy of QuickMove and the DeskTop should be kept in each partition that you use under GEOS. On systems without ram expansion, take this opportunity to move a copy of CONFIGURE into each partition, as well. (Note: on ram expanded systems this last step is unnecessary.)

In order to make moving these three files into each partition easier, copy them first into one partition using the procedures described above. Then enter QuickMove, and use its multiple file copy procedure (described below in Using QuickMove) to copy the files into other partitions which you intend to use with GEOS.

There is also a hidden advantage for those CMD HD owners who also own Commodore's 1581 disk drives- increased performance! Special serial routines have been used to increase the byte transfer speed of the 1581 by 75% over the original Berkeley driver.

Using QuickMove

QuickMove is the pivot of the CMD HD/GEOS environment In addition to allowing easy access to other 1581 partitions, QuickMove enables multiple file coping between partitions. Up to 9 files can be selected for copying into another partition.

Opening QuickMove is as simple as double clicking on its icon. After a moment the application screen opens and QuickMove begins to fetch partition information from the HD. On ram expanded systems, this partition info is then stashed to an unused portion of the REU.

When the application screen first opens, the current partition is declared in a box at the top of the application's screen. At the base of the screen a graphic of the CMD HD indicates drive activity by flashing the ACTIVITY light. A short menu offers you two choices: geos and options.

To facilitate moving between partitions, access to the first 9 1581 partitions is provided through the number keys. It's important to understand that pressing 9, for example, does not take you to partition 9. Rather, it will take you to the ninth 1581 partition on your HD.

Once you have moved to the new partition (the Current Partition declared at the top of the screen will reflect the move), press C= Q to exit back out to the DeskTop.

Using the Options Menu

The options menu provides access to the real power of QuickMove: The ability to view the directory of other partitions and to copy files between them. Clicking on the Options selection unfolds another menu revealing 3 additional choices: `open partition`, `show directory`, and `quit`.

Open Partition C= P

Clicking on this option, or alternatively pressing `C= P` brings up a dialog box in the center of the screen. The first 7 1581 partitions are presented. An arrow points to the first selection in the box. To select another partition, simply click on its name or use the CRSR DWN key to move the arrow to the desired partition. Press RETURN or click on `OKAY` to move to that partition. The Current Partition box is updated to reflect the change.

Should you wish to back out of this process, simply click on `CANCEL`, or press `C= Q`. You will be taken back to the QuickMove application screen.

Entering another partition through the Open Partition option automatically moves you into the show directory option.

Show Directory C= D

Selecting this option brings up a dialog box displaying the first seven files of the directory. Six icons below the file display window allow easy movement within the directory. From left to right, they allow you to:

- move to the bottom of the directory.
- move to the top of the directory.
- scroll the list up one filename.
- scroll the list down one filename.
- move the list down one page (7 entries).
- move the list up one page.

An arrow points to the currently selected entry. The arrow is manipulated in the same fashion as above. Clicking on a filename will move the arrow to that filename. The CRSR DWN key may be used as well.

Assuming that there are files in the directory you are viewing, two gadgets appear on the right side of the dialogue box: COPY and COPY TO. To select a file for copying simply click on its name or press `C= C`. Above the COPY gadget, the words `1 file(s) selected` will appear. Up to 9 files can be selected in this fashion. In each instance, the number of files selected will increment Currently, there is no option to deselect a file.

To select the destination partition, click on `COPY` to or press `C= K`. The Open Partition dialog box will appear. Select the destination partition using the same procedure described above. Once you click on okay, the copying process will begin. As the copy operation progresses, the Hie currently being copied is displayed at the top of the screen.

If the destination partition is not a GEOS partition, you will be asked whether you wish to make it a GEOS partition. Clicking on `YES` will write out the offpage directory block and GEOS format string to the destination. Clicking on `NO` will abort that request In either case, the copy operation proceeds immediately afterward.

If the file already exists at the destination, QuickMove will ask whether it is okay to overwrite the file. Clicking on `NO` aborts the copying of that file. If more files remain in the copy queue, the procedure moves on to the next file. Clicking on `YES` deletes the old file, and replaces it with the new one.

When the copy process is complete, QuickMove returns to the initial screen. The destination partition is now the current partition.

Booting GEOS from the HD

It is possible to boot GEOS from a 1581 partition on the HD. This requires a utility which is capable of making a 1581 GEOS boot disk, such as the Maverick V3 or V4 utility packages which were distributed by Software Support International (SSI).

There are two ways of creating a GEOS boot partition. Before trying either method, follow all the steps given previously in this section to create a working GEOS system disk with the special HD files on it. This must be done on an original GEOS system disk- not on a copy or a backup.

After you have your original system disk working properly with the HD, you may use the 1581 GEOS boot disk utility program to create a bootable partition. We have done this successfully using Maverick V3 and V4.

The following procedure describes the steps you must follow to get GEOS to boot from your HD:

1. Power on your system.
2. Use the HD's Change Partition command to place the HD into the 1581 partition you have chosen for your GEOS boot partition.
3. Use the SWAP 9 switch (or the software command) to make it device number9.
4. Boot your utility program for creating a 1581 GEOS boot disk on your floppy drive (device number 8).
5. Follow the steps given with your utility program to create the 1581 GEOS boot disk, using the HD as the target 1581.
6. Once the process is complete, boot GEOS once again from your regular boot disk, then copy the HD specific files (CONFIGURE, HDTime, and QuickMove) to your GEOS boot partition on the HD.

The next time you want to boot GEOS, you should be able to turn on your system, change the HD device number to 8 (with SWAP 8), change to the proper partition (your GEOS boot partition), then LOAD and RUN the boot program as described in the documentation provided with the utility you used to create the GEOS boot files. If you are using GEOS 128, and you used Maverick to create the boot partition, type BOOT from within that partition and then press RETURN to start GEOS.

Maverick V5 contains routines which will not allow this method to be done directly on the HD. In this case you must create the 1581 GEOS boot disk on an actual 1581 disk drive. After you have done this, use MCOPY to copy that disk to the proper partition on the HD. Again, you must create that 1581 GEOS boot disk from an original system disk which has been modified to work with the HD. After you have completed copying the 1581 disk to the proper partition on the HD, proceed to Step 6 in the above instructions.

GEOS HD Do's and Don'ts

Do:

- Copy your GEOS applications over to the HD. It is 100% compatible with all GEOS applications.
- Give us your suggestions. Drop us a postcard. Let us know how its going and how we, at CMD, can make it better.

Don't:

- Never press the HD's RESET button once inside GEOS. At the very least, you will lose the HD from the system until the next reboot.
- For the same reason, don't press either of the SWAP buttons after starting GEOS (WRITE PROTECT may be used freely).
- Don't call Berkeley Softworks if you have a problem with the HD - call us. CMD, not Berkeley Softworks, is responsible for compatibility between the HD and GEOS. If you experience any problems while using the HD with GEOS, power down and reboot with the HD disconnected. If that solves the problem, contact us immediately. We will be glad to help solve your problem. Please, do not request assistance from Berkeley Softworks with HD-related problems.

Section 6
Using CP/M

Operating with CP/M

The CMD HD hard drive currently supports CP/M PLUS V3.0. The version we have tested is release date 28 May 1987. Earlier versions should work with 1541 and 1571 partitions, although they have not yet been tested. Currently, CP/M will work with the following types of partitions using the formats given in the table below:

1541	-	C128 Single Sided
1571	-	C128 Double Sided
1581 CP/M	-	Commodore 1581 (Double Sided)

Installation of CP/M

As is indicated in the above list of supported formats, using the CMD HD with CP/M is currently limited to Commodore disk emulation. The amount of storage space available under CP/M is enhanced by use of the file SPORT.COM (located on the HD CP/M UTILITIES disk) which allows you to move between different partitions. You may have as many CP/M partitions as you wish, but no partition may be larger than a 1581.

The following instructions will allow you to create a bootable version of CP/M on the HD. This may be done in any one of the partition types listed above, although we do recommend using 1581 CP/M partitions for the highest capacity per partition.

1. If the HD is not configured as device number 9,10, or 11, you should first configure it as such with HD-TOOLS or use the SWAP 9 switch to set it as device number 9.

2. Before booting your CP/M PLUS V3.0 system disk, place the hard drive into the partition you wish to use for CP/M by using the Change Partition command.

3. Insert your CP/M PLUS V3.0 system disk into the floppy disk drive device number 8 and boot the system as you normally would. This can be done by typing:

   ```
   BOOT
   ```

   and then pressing the RETURN key.

4. At the A> prompt, issue the FORMAT command and press the RETURN key.

5. After the FORMAT.COM file has loaded, a menu will be displayed. Using the cursor keys located at the top of your keyboard, move the selection bar on the screen until 'Select Different Drive' is highlighted, then press RETURN.

6. Now you will need to select the drive designator for the HD. Drive A is device number 8, B is number 9, C is 10, and D is device number 11. Cursor to the correct selection for the HD's current device number and press the RETURN key.

7. CP/M will now inform you which drive type is selected, and present you with choices for format types. Choose the format type according to the list given at the beginning of this section. If you are using a 1581 CP/M partition, you must use the full format- writing only the system track will not work properly on the HD. Press RETURN when the proper format is highlighted.

8. CP/M will now inform you to insert the diskette to be formatted into the drive. This is unnecessary since the HD's disk is not removable. When you are ready to format the partition, hold down the SHIFT key and press the number key marked 4. Pressing any other key will abort the process.

   CP/M will now format the partition. It is not unusual for CP/M to leave a file open on the drive after this process is complete if you have selected a 1541 partition. This will be indicated by the upper activity lamp on the HD remaining lit

If you wish to place a copy of the CP/M operating system on the HD, continue with the following steps. Otherwise, you may begin using this partition with the HD. Be sure to place a copy of the SPORT.COM program in this partition if you plan to use more than one partition with CP/M. Directions for doing so are given later in this section.

9. Copy the CP/M system file from your boot disk to the HD partition using the PIP.COM program. The following example shows how this is done, but be sure to use the proper drive identifier for the HD. The example below would be used if the HD is set as device number 9 (indicated by the letter B). Change B to C if your HD is device number 10, or change B to D if it is device number 11.

```
A>PIP B:=A:CPM+.SYS
```


10. Copy the CP/M command file CCP.COM to the HD. Again, the example given will show you how this is done if the HD is device number 9. Make the appropriate change to the drive identifier if your HD is set for some other device number.

```
A>PIP B:=A:CCP.COM
```

    You could instead copy all COM files to the HD by issuing:

```
A>PIP B:=A:*.C0M
```

This is all that is required to create a bootable version of CP/M on the HD. You may now boot CP/M from the HD if you set the HD as device number 8 with HD-TOOLS or by using the SWAP 8 switch.

Changing Partitions in CP/M

You may move among different partitions on the HD using the program SPORT.COM. This file is located on the CP/M UTILITIES disk which came with your HD. Copy this file to each partition you use for CP/M by using PIP. This can be done more easily if you place a copy of PIP on the CP/M UTILITIES disk first.

SPORT.COM allows you to send Commodore-style DOS commands to any serial port device. SPORT is issued as a command with up to three command parameters. These parameters are assigned values for the device number, secondary address and command. If issued without any parameters, SPORT will present a help screen. Here is the syntax:

```
SPORT[.COM] [/Ddv] [/Ssa] [/C'command')
```

where: dv = The device number (in hexadecimal) of the target drive
sa = the secondary address (in hexidecimal) of the command channel (15)
command = the command string (be sure to enclose the string within single quotes.)

SPORT works in a manner similar to most DOS wedges. Here is an example of how to change partitions on an HD set as device number 9:

```
SPORT /D9 /SF /C'CP5'
```

This command makes partition number 5 active on the HD. Once you have entered a new partition, you should use the Control-C key combination to log in the new partition. For example, if you are currently on drive A (the last prompt was A>) and the HD is drive B, type the letter 'B', a colon (:) and then press RETURN. The next prompt you get should be the drive B prompt. Now hold down the <CONTROL> key and press the letter 'C'. The partition is now logged and will be handled properly. Failing to log in partitions may result in undesirable operation.

Section 7
Using Q-Link

About Q-Link

Q-Link is an inexpensive nation-wide electronic network service provided by Quantum Computer Services for use by Commodore computer owners. Among other things, Q-Link provides its members with the following services:

- INFORMATION - News, online encyclopedia, seminars, workshops
- SOFTWARE - One of the largest collections of Commodore programs
- ONLINE GAMES - Play by yourself and with other members
- CHAT AREAS - Where you can converse with other users and friends
- SUPPORT - A direct link with CMD and other companies
- MESSAGES - Classified ads and special interest areas

There are many, many more features on Q-Link. If you are not already receiving the benefits of a low cost membership on this service exclusively for Commodore users, sign on by using the provided software. There's a lot of information just waiting for you, and it's also one of the fastest ways to get help or advice about using your CMD products.

Signing on to Q-Link

If you have never been on Q-Link before, you will need some information. This information is provided in the Q-Link Kit which accompanied your HD. If the Q-Link package was not provided for some reason, contact Quantum Computer Customer Service at 1-703-883-0788, or if you are calling from Canada 1-703-883-0789.

Everything you need to know to log-on for the first time is described in the Q-Link brochure. However, make note of the following differences given below. If you have problems logging on for the first time, contact Q-Link at 1-800-392-8200 (US) or 1-703-883-0789 (Canada).

The CMD/Q-Link System Disk: The software contained on this disk is already located in partition 6 of your HD. You can (and should) use MCOPY to make a backup of this software.

Loading Q-Link from the HD

There are some differences to note when loading the special HD-compatible Q-Link software from your HD. Follow these steps when loading the Q-Link software from the HD for the first time:

1. Turn the power on to your monitor, CMD HD, and computer. If you are using a Commodore 128 or 128D, hold down the COMMODORE key while turning on the computer so that it will be in 64 Mode.

2. Partition 1 contains a program called BOOTQ. Type the following:

   ```
   LOAD"1//:BOOTQ",12
   ```

   and press the RETURN key. If you have changed your HD's default device number, substitute the '12' at the end of the LOAD command with the correct device number.

3. When the READY prompt appears, type:

   ```
   RUN
   ```

   and press the RETURN key.

If the Q-Link software was not pre-installed on your HD

If the special HD version of the Q-Link software did not come pre-installed on your HD, copy it to a 1541 Emulation Mode partition by using MCOPY. After you have copied the Q-Link disk to the HD, turn your computer off and back on again. Change to the partition where you placed the Q-Link software by using the CP command. Next, press the SWAP 8 button, and type:

```
LOAD"*",8,1
```

and press the RETURN key.

If you can, use partition 6 for the Q-Link software. You may then copy the BOOTQ program from the HD UTILITIES disk into partition 1 using FCOPY. This will allow you to use the same instructions as those who received the Q-Link software already installed on the HD.

Note: If you have problems with disk errors on the HD or the CMD/Q-Link System disk provided with the HD, contact CMD for assistance. CMD, not Quantum Computer, is responsible for any software supplied on or with the HD.

Q-Link Members and Backups

If you arc already a member of Q-Link, use the program located on your HD UTILITIES disk called XFER QDATA. This program will allow you to transfer your Q-Link sign-on data to the HD. Upgraders: this should be done after placing a copy of the Q-Link software on your HD.

The program XFER QDATA may also be used for backing up your Q-Link sign-on data to a floppy disk after each session. That way, if anything should ever go wrong with your HD, you could still get on Q-Link without having to call (hem for information. You should back up your sign-on data to a copy of the Q-Link system disk or to a floppy disk copy of the Q-Link partition (created with MCOPY).

Uploading, Downloading, and Other Disks

You may use other partitions on your HD for uploading and downloading software while connected to Q-Link. To do this, study the instructions for using FPPS Mode, located elsewhere in this manual. When you wish to upload or download a file, move to the appropriate partition using FPPS Mode. This should be done just prior to selecting the 'Upload a file' or 'Download this file' from the Q-Link menu. You should only use 1541 Emulation Mode partitions for doing this, unless you have created or are using a patch for one of the other partition types. The patch for 1581 disk drives located on Q-Link will work with 1581 partitions. The 1541SUB program can also create a suitable patch in an empty Native Mode partition.

You may also use CASINO and SUPER-Q with your HD. These should be placed in 1541 Emulation Mode partitions, and can be accessed with the FPPS Mode as well

CLUB CARIBE and PUZZLER may not be used on the HD. It is possible to upload, download, or use CASINO, PUZZLER or SUPER-Q on a floppy disk by using the SWAP 8 button. This will cause the floppy disk on your system which is normally device number 8 to be used.

Note: Be sure to change back to your main Q-Link partition, or SWAP 8 back to the HD (if you have SWAPed to another drive) whenever the system requests you to insert your Q-Link disk.

To use CLUB CARIBE it will be necessary to boot with a standard Q-Link boot disk on a floppy disk drive, and may also require disconnecting the HD and other devices attached to the serial bus. For further information on using CLUB CARIBE, see the Club Caribe Users Guide.

Section 8
Using Commercial Software

Types of Commercial Software

There are many types of commercial programs, but for our discussion here we will separate these programs into two main categories- those which are copy-protected and those which are not Most commercial programs, with the exception of games, are compatible with the HD in one form or another. This section of the manual will help you to discover which programs will work with the HD and the steps required to get them working.

Software without copy-protection

Software which is not copy-protected can almost always be installed directly on the HD. The main question usually concerns the type of partition to use.

The best way to determine which partition type to use is to experiment, starting with a Native Mode partition. It is usually best to start by creating a Native Mode subdirectory to hold the files used by the particular program. Use the HD's Make Directory command to do this and then copy all the files from the program disk to that subdirectory by using FCOPY.

To test whether the program works or not, change to the partition and directory where you have placed the program and begin running it as you would from a floppy disk. It may be necessary to set the HD as device 8 or 9 by using the SWAP buttons on the front panel (most software requires device number 8). If the software does not work from within a subdirectory, try using the root (or main) directory in a Native Mode partition.

Some software, even though it contains no copy-protection, will only work with a certain type of drive. Such programs will usually work in one of the Emulation Mode partitions on the HD. Most often, the partition will be a 1541 or a 1571 Emulation Mode partition, though many programs will work in a 1581 Emulation Mode partition as well.

When trying Emulation Mode partitions, work your way 'downward', trying a 1581 partition first, then a 1571, and finally a 1541 partition. Before testing, be sure to use the Change Partition command to move to the partition in which you placed the program. If the software will still not work in any of the partitions, all is not lost, as there may be some 'hidden' data on the program disk which cannot be duplicated with FCOPY. In this case, use MCOPY to copy the entire disk to a partition of the same type. If it still will not work, the software is probably performing some very drive specific tasks. This is rare unless the program is some kind of disk utility.

Multiple Disk Programs

Some programs are distributed on a number of disks. In many cases, simply copying all the files from each of the disks into a single partition using FCOPY will work, and this is the first method to try. Watch out for file names which are the same when copying these types of disks. These files may or may not be identical. As long as a multiple disk program uses standard files and as long as it can determine that the proper disk is in use while searching for a Tile, this method will work.

If some of the information for a program is stored directly on the disk without a file name, or the disk name is checked to determine if the correct disk is being used, it will be necessary to use separate partitions for each of the disks. These partitions will have to be of the same type as the original program disk, and you will need to use MCOPY to copy the disks to the HD partitions.

There may also be times when a combination of methods can be used. For instance, a word processor which comes on four disk sides may contain standard files on three of the disk sides while the fourth side contains a dictionary or thesaurus stored in a special manner on the disk. In this case, it is likely that you could combine the first three program disks into a single Native Mode partition or subdirectory, and then copy the fourth disk to a partition of the same type as the original disk. The files from the first three sides could be copied with FCOPY, with the remaining side copied using MCOPY.

If a program must be stored in different partitions, you can use FPPS (Front Panel Partition Selection) on the HD to change partitions when the program requires you to do so. Other programs allow you to send disk commands, and in such cases you could use the disk command option to send the CP (Change Partition) command to the HD.

Other Solutions

Some software expects to find the disk directory in a certain place on the disk. Normally this kind of software can be operated using an Emulation Mode partition of the required type. It may also be possible for this software to be operated in a Native Mode subdirectory which has been specially created to simulate the directory of a 1541 or 1581 disk drive. These types of subdirectories can be created in an empty Native Mode partition which has the required number of tracks by using the `1541SUB` and `1581SUB` utilities supplied with the HD.

Creating an emulation subdirectory is the method used to extend the amount of usable disk space with programs such as Superbase. Be aware, however, that some programs (such as Superbase) check to see how many tracks or sectors it can access and determines the type of drive it is using by doing so.

As a result, it may be difficult to determine which subdirectory type to emulate (1541 or 1581). As long as you use Native Mode partitions with 40 or more tracks, 1581 emulation subdirectories will usually work with these types of programs. Other programs will always assume a 1541 no matter how many tracks you have in the current partition.

Whenever you use a subdirectory for emulation purposes, remember to use the Change Directory command to make that directory the current one before using your software. This may be done from another partition, as long as you include the proper partition number and path in the command.

You may encounter one other situation with software that is not copy-protected: built-in fast loaders (usually for 1541 disk drives). Since fast loaders are written in a drive specific manner, programs which do not allow you to turn the fast-loading option off, or which do not check to make sure that a 1541 is being used will not load properly from the HD.

Copy-Protected Software

Normally, copy-protected of software cannot be placed directly onto the HD. You may, however, be able copy some of the more mildly protected programs by using MCOPY. This tends to work more often with software supplied on 1581 disks than it does with 1541 or 1571 formatted disks.

If a program will not load from the HD, you may be able file copy most of the data files to the HD with FCOPY, and then load the program initially from the floppy disk. After the program stops loading (and is past checking the copy-protection) you could press the SWAP 8 button to substitute the HD for the floppy. Any subsequent disk accesses will be directed to the HD.

Some programs do not access the disk once they have loaded. If a program is copy-protected and has no need of a disk drive after it has been loaded, there is not much chance of using it with the HD. The only method left to try with this type of software is use a memory capture type of cartridge to save the program as an unprotected file. There are also some utilities (such as Maverick) which will make copies of software with the protection removed by using a parameter disk. This too is a good method for making HD bootable copies of protected software.

Section 9
Command Reference

Command Syntax

This section documents many of the commands which can be used with the HD. The syntax for these commands is given in a standard format which should allow you to easily recognize required and optional parameters. Examples are used throughout to assist you in determining proper usage of the commands. If any problem should arise in determining command syntax, be sure to check the following information.

Command String Elements

The command string is made up of a number of elements. In the case of commands sent directly from BASIC, the first part of the command string is usually the command itself. In the case of commands sent via the disk drive command channel, the command itself is usually found at the beginning of a string sent to the disk drive. The elements of the command string as used in this manual are described below:

Literals are characters which must be entered exactly as shown. These will appear as plain text

User supplied values are those which must be supplied by the user and whose values and type are dependent upon the use of the command. These will appear as *italicized* text.

Optional parameters and options are values or literals which need not be included in the command unless the user wishes to specify the option. Often, the optional parameters will be substituted with a certain default when left out of the command string. Optional parameters will appear within [brackets].

Choice parameters allow or require you to choose from more than one parameter to be placed within the command string. Whenever these appear in the syntax of the statement, the choices will be enclosed by {braces} and the individual choices will be separated by a vertical bar character (|). Only one choice parameter may be used in command string.

You may occasionally notice syntax in which one or more elements is followed by three periods (...). This means that the parameter last shown may be repeated. This will normally be discussed in more detail in the text describing use of the command.

Example command string

The following example illustrates the command syntax used throughout this manual:

```
HEADER"partname"[,Iid][,Dn][{ON|,}Udv]
```

In this command string, HEADER is the actual command and is entered literally. The commas (,) and the 'I', 'D', and 'U' are literals contained within brackets and are optional. The parameters 'partname', id, n, and dv are user supplied variables. These variables are always described in a table following the command syntax. The table for this command string would look like this:

where:	partname	=	the name you wish to appear in the partition header
	id	=	a two character id for the partition header
	n	=	the partition number you wish to format (0 or 1)
	dv	=	the device number of the HD

Since these parameters are all enclosed in brackets, they are optional. If you decide to use one or more of the optional parameters, notice that they are accompanied by a literal. This literal must be included in the order shown along with the user-supplied variable. Also notice that the 'ON' and a comma are enclosed in braces and are separated by a vertical bar. This means that if you want to supply the unit number, it must be preceded by either the literal 'ON' or by a comma. The full use of this command is shown in the following command string:

```
HEADER"PARTITION 1",IP1,D0,U12
```

Since partition (or drive) zero (0:) is assumed in BASIC 7.0 commands, It would also be possible to shorten this command to:

```
HEADER"PARTITION 1",IP1,U12
```

Paths in Command Strings

Throughout this manual you will see commands which have [path] shown in the command syntax. Paths are only used when accessing Native Mode partitions, and specify which subdirectory the disk operation is intended for. Here is an example of a command with a path in its syntax:

```
VERIFY"[[n][path]:]filename",dv[,sa]
```

Most of the common commands used on the HD will allow you to include a subdirectory path within the command string, provided that the target file is within a Native Mode partition. This path immediately follows the partition number within the command string and is shown throughout this manual within command syntax descriptions as [path]. When including one or more subdirectories within a command string, each subdirectory placed in the

Command must be bracketed between slash (/) characters, and the final slash must usually be followed by a colon.

For example, if you had a file named COPY located in a subdirectory named UTILITIES in partition number 1, you could load this file with the following command:

```
LOAD"1/UTILITIES/:COPY",12
```

The portion of this command which makes up the path is:

```
/UTILITIES/
```

If you had nested subdirectories, and the file COPY was located in a subdirectory named COPIERS which in turn was located within as subdirectory named UTILITIES in partition 1, you might load this file with:

```
LOAD"1/UTILITIES/COPIERS/:COPY",12
```

The portion of this command which makes up the path is:

```
/UTILITIES/COPIERS/
```

You may be able to shorten this command, depending on which directory and partition you are currently located in. For example, if you are already located within partition 1, and the root directory is your current directory, you could skip the partition number in the command string:

```
LOAD'/UTILITIES/COPIERS/:COPY",12
```

If your current partition is partition 1 and your current directory is UTILITIES, you could simply enter:

```
LOAD"/COPIERS/:COPY",12
```

If your current partition is partition 2 but your current directory in partition 1 is UTILITIES, you could enter:

```
LOAD"1/COPIERS/:COPY",12
```

If your current partition is partition 2 but the current directory in partition 1 is COPIERS, you could enter:

```
L0AD"1:C0PY",12
```

There is another syntax which will always allow you to begin your path at the root directory. This is helpful when you are located within a different subdirectory in the same partition. For example, if your current directory is GAMES in partition 1, and you wish to load the COPY program shown in the previous example, you can begin your path with two slashes:

```
LOAD"//UTILITIES/COPIERS/:COPY",12
```

Two slashes placed at the beginning of a subdirectory path indicates that the path must begin at the root directory. If you are in a different partition, or are not sure which directory is the current directory in the partition you wish to access, it is usually wise to use the double-slash method. Remember to include the partition number if you are in a different partition:

```
LOAD"1//UTILITIES/COPIERS/:COPY",12
```

Subdirectory Paths Using JiffyDOS Commands

The examples given above are shown using the standard Commodore DOS methods of loading files. If your computer is equipped with JiffyDOS, you may use the JiffyDOS load commands. Here are the above examples converted to their JiffyDOS equivalents:

```
/"1/UTILITIES/:COPY",12
/"1/UTILITIES/COPIERS/:COPY",12
/"/UTILITIES/COPIERS/:COPY",12
/"/COPIERS/:COPY",12
///UTILITIES/COPIERS/:COPY
/1//UTILITIES/COPIERS/:COPY
```

You may also leave out the quotes (") and the comma and device number by using the JiffyDOS <CONTROL><D> function to change the default device. The last two examples illustrate this option and show the only situation under which you will use three slash characters in a row. The first slash character is assumed to be the BASIC LOAD command by JiffyDOS. You may also use any of the other JiffyDOS wedge commands for loading and saving programs in conjunction with subdirectory paths. See your JiffyDOS manual for more information about these commands.

Sending Commands from Basic

Most of the commands you send to the HD will be from BASIC. This requires very little new knowledge since the HD accepts standard disk drive commands. You should note that in order to maximize your use of the HD, it may be desirable to use BASIC 2.0 or DOS command channel commands when operating the HD on a C128 instead of using BASIC 7.0 commands. This is because BASIC 7.0 places some limitations on the use of partition numbers in its command syntax. Specifically, the drive number parameter in BASIC 7.0 commands may only be represented by a zero (0) or a one (1). Since the HD uses this number as an indication of which partition is to be used for the particular command, and since the HD can have up to 254 partitions, the BASIC 7.0 commands may or may not be able to access the desired partition. Also, because of the way in which BASIC 7.0 sends commands, it is not possible to include subdirectory paths within these commands. If you wish to include subdirectory paths, use the BASIC 2.0 or DOS command channel version of the command instead.

The Command Channel

Many of the commands discussed in this section, require you to send the command via the disk drive's command channel. Opening a channel to the drive requires the following BASIC statement:

```
OPENlf,dv,sa
```

where:

`lf`	=	the logical file number
`dv`	=	the device number of the drive
`sa`	=	the secondary address

The logical file number can be any number from 1 to 127. Other numbers are legal (128-255) but will cause side effects which are not usually desirable, so it is wise to avoid them.

The device number is the same as the device number currently assigned to the HD. This is set to 12 at the factory, but may be changed to any number from 8 to 29 by using HD-TOOLS, or set to 8 or 9 by using the SWAP feature.

The secondary address is often referred to as a channel. Secondary addresses 0 through 14 are used to open files, whereas secondary address 15 tells the drive that data sent via this channel should be interpreted as commands and command data. Thus, channel 15 is referred to as the 'command channel'. This example shows how to open the command channel:

```
OPEN15,12,15
```

The command channel may be opened from within a program or in 'direct' mode. Whenever you enter a command to be executed immediately, without preceding it with a line number, it is considered to be entered in BASIC'S direct mode. Here is an example of sending a command via the command channel:

```
OPEN15,12,15:PRINT#15,"I":CLOSE15
```

This type of command may also be sent without using the PRINT# command as shown below:

```
OPEN15,12,15,"I":CLOSE15
```

Some commands require that other parameters be sent to the drive in the form of character strings with the CHR$ function. These types of commands can be sent using the PRINT# command. An example of this is the device number change command:

```
OPEN15,12,15:PRINT#15,"U0>";CHR$(10):CLOSE15
```

The semi-colon which appears between the portion of the command in quotes and the character string code is optional. Some commands require that numeric variables or actual numbers be used as command parameters. This is more common with direct access commands. An example is given in the following Ul command:

```
OPEN15,12,15:PRINT#15,"Ul";2;0;1;34:CLOSE15
```

The numbers shown following the semi-colons in the command given above could also be expressed as variables. For example:

```
OPEN15,12,15:PRINT#15,"Ul";C;D;T;S:CLOSE15
```

If numbers are used in a command, they may be included within the string portion as long as each is separated from the string and the other numbers by a space:

```
OPEN15,12,15:PRINT#15,"U1 2 0 1 34":CLOSE15
```

In this case, the trailing quote is placed after the last number or parameter required by the command. You may optionally place a colon at the end of the command itself, no matter which way the command is used:

```
OPEN15,12,15:PRINT#15,"U1:";2;0;1;34:CLOSE15
OPEN15,12,15:PRINT#15,"Ul: 2 0 1 34":CLOSE15
```

Reading Disk Errors

Disk errors are most often detected by reading the command channel. This is done by using either the GET# or INPUT# commands. Using INPUT# is the fastest method of returning error information from the drive. GET# is more commonly used for getting non-error information. Since both the GET# and INPUT# commands must use the BASIC input buffer, they cannot be used in direct mode.

Usually, programs will check for errors immediately after attempting to perform a disk access or disk command. Here is a short program which shows how the error channel is read:

```
10 OPEN15,12,15:INPUT#15,E,E$,T,S:CLOSE15
20 PRINTE,E$,T,S
```

As you can see, four parameters are returned via the command channel when checking for an error. These are, in order the error number, the error message, the track where the error occurred, and the sector where the error occurred. Many particular errors will not occur at any particular track and sector, in which case the track and sector variables will contain zeroes.

There are occasions where it is more desirable to obtain error data or other information from the command channel one byte at a time. In these instances a program similar to the one that follows could be employed:

```
10 OPEN15,12,15
20 GET#15,E$:PRINTE$;:IFST<>64THEN20
30 CLOSE15
```

If you are using JiffyDOS, it is possible to read the error channel without using a program. This is done by pressing the commercial at (@) key and then<RETURN>.

In the preceding example, we used the status variable to determine when the end of the file was reached. The BASIC status variable ST is quite useful in serial device access, so here is a breakdown of the individual bit values. These definitions apply when the specified bit is set, or equal to one.

BIT	DESCRIPTION OF STATUS	STATUS VALUE
7	DEVICE NOT PRESENT or END OF TAPE	128
6	END OF FILE (EOI)	64
5	CASSETTE CHECKSUM ERROR	32
4	VERIFY ERROR or CASSETTE READ ERROR	16
3	DATA BLOCK TOO LONG (Cassette)	8
2	DATA BLOCK TOO SHORT (Cassette)	4
1	TIME OUT ON LISTENER	2
0	TIME OUT ON TALKER	1

Figure10–1

The C128 provides another method of checking for disk errors via two reserved BASIC7.0 variables: DS and DSS. The variable DS returns the error number, while DSS returns the error message string. These variables can be viewed with a PRINT statement, as shown in the following example:

```
PRINTDS,DS$
```

It is not necessary to print both of these variables at the same time, but the error message string will usually provide help in understanding why the error occurred, and also makes it easier to lookup in the error descriptions located in Appendix B of this manual.

Note: DS and DSS are usually valid only after a BASIC 7.0 disk command.

Partition Numbers in File Names

A partition number may be specified within a filename in place of the drive number. The drive number is the number which can precede the colon (:) in Commodore DOS filenames. Often, this number is not included since it is normally used only with dual drives to determine if the file operation should be directed to drive 0 or drive 1. Whenever this drive number is left out of the command, drive 0 is assumed. These rules also apply to the HD, which instead of containing 1 or 2 drives, contains a single drive with up to 254 partitions. Whenever you wish to perform a command or file operation with the current partition, you may use a 0 or leave out the partition number entirely. However, if your current partition is different than the partition in which you wish to perform the file operation, you must either move to that partition first (with the 'CP' command), or specify the partition number within the filename. The following examples should help to illustrate this:

```
LOAD "1:MC0PY",12
OPEN2,12,2,"3:TESTFILE,S,W"
```

Load commands may be abbreviated if you are using JiffyDOS:

```
/1:MCOPY
```

Partition Numbers in Disk Commands

As described above, partition numbers may be used to replace the drive number in filenames. Partition numbers may also be used in this manner when sending disk commands. In fact, any disk command which allows inclusion of a drive number (with the exception of direct access commands), will also allow you to substitute a partition number in its place. You may use partition numbers when formatting, copying, renaming, scratching, validating, and initializing. This is a large part of what makes the HD so compatible with Commodore DOS.

In the case of direct access commands, the current partition is assigned to the direct access file when that file is opened. As a result, you must use the 'Change Partition' command to select the desired partition before opening a direct access file. Once the file has been opened, all commands sent to that direct access file will refer to the partition you were in when the file was opened, even if the current partition is changed.

Partition Commands

Many of the commands used on the HD are partition related and, among other things, are used to format, initialize and change partitions.

Creating Partitions

Partitions are created by using the HD-TOOLS program supplied with the HD. Use of this program is documented in Appendix A.

Creating 1581 Style Sub-partitions

You can create 1581-style partitions within 1581 Emulation Mode partitions on the HD. This type of partition is sometimes referred to as a subdirectory by 1581 users. Because of the way these partitions allocate space, we feel the term subdirectory does not apply. Since the HD is already divided into partitions, and since these 1581 'subdirectories' are nested into 1581 Emulation Mode partitions, we have opted to call them sub-partitions.

Certain limitations apply when creating sub-partitions that are intended to store files. Because of the way physical tracks are handled in Commodore's 1581 DOS, and because sub-partitions must contain header and directory blocks, the minimum size of a sub-partition is 120 blocks. The starting sector must be zero, and the ending sector must be a multiple of 40. Sub-partitions are not allowed to begin on, end on, or contain within themselves track 40. Here is the syntax required to create 1581 style sub-partitions:

```
OPEN lf,dv,15:PRINT#lf,"/[n] :partname,"CHR$(st)
CHR$(ss)CHR$(sl)CHR$(sh)",C":CLOSE15
```

where:
- lf = the logical file number for the command channel
- dv = the current device number assigned to the HD
- n = the target 1581 Emulation Mode partition
- partname = the name of the 1581 sub-partition name to be created
- st = the starting track of the sub-partition
- ss = the starting sector of the sub-partition
- sl = the low byte of the sub-partition size in sectors
- sh = the high byte of the sub-partition size in sectors

Example:

```
OPEN15,12,15:PRINT#15,"/4:SUB1,"CHR$(1)CHR$(0)
CHR$(160)CHR$(0)",C":CLOSE15
```

The preceding example should be entered as one line. There is no JiffyDOS equivalent for this command since it requires the use of the CHR$ function.

> Note: Before you can use a newly-created 1581 sub-partition, you must format it. See 'Formatting 1581 Style Sub-Partitions' for more information.

Deleting Partitions

Partitions are deleted by using the HD-TOOLS program supplied with the HD. Use of this program is documented in Appendix A.

Deleting 1581 Style Sub-partitions

1581 style sub-partitions are handled quite differently than standard HD partitions. You can delete a sub-partition by using the DOS or BASIC 7.0 SCRATCH commands. Scratching a 1581 sub-partition is no different than scratching a file, although the consequences may be severe if this is done accidentally. Any files contained within the sub-partition will be lost See 'Scratching (deleting) Files1 for the proper command syntax. Remember to substitute the name of the sub-partition for the filename.

Changing Partitions

You may change from one partition to another by sending the 'CP1 (Change Partition) command to the HD via the command channel. The syntax is:

```
OPENlf,dv,15: PRINT#lf,"CPn":CLOSElf
```

where: lf = the logical file number for the command channel
dv = the device number of the HD
n = the partition number you wish to change to (1-254)

Example:

```
OPEN15,12,15:PRINT#15,"CP4":CLOSE15
```

JiffyDOS Example:

```
@CP11
```

The 'C'<SHIFT>'P' command is a variation on the Change Partition command which allows it to be used more easily from within BASIC programs. A shifted 'P' is indicated by the symbol '□' This command allows you to use a character string to indicate the partition. The syntax is:

```
OPENlf,dv,15: PRINT#lf,"C□"+CHR$(n):CLOSE
```

where: lf = the logical file number for the command channel
dv = the device number of the HD
n = the partition number you wish to change to (1-254)

Example:

```
OPEN15,12,15:PRINT#15,"C□"CHR$(11):CLOSE15
```


Moving Between 1581 Style Sub-partitions

Since the 1581 Emulation Mode partitions on the HD support 1S81 style partitioning (that is to say 'sub-partitioning'), you may use the standard DOS commands to change from one 'sub-partition' to another. The syntax is:

```
OPENlf,dv,15:PRINT#lf,"/[n]:[partname]":CLOSElf
```

where: lf = the logical file number for the command channel
dv = the device number of the HD
n = the partition number of the 1581 Emulation partition
partname = the partition name of the 1581 style 'sub-partition'

If the 1581 Emulation Mode partition is the current partition, the partition number n may be left out The 'sub-partitions' may be nested within each other, but to access one which is two levels down from the currently selected one, you will have to issue the command twice (once with each of the two sub-partition names). To return to the root (main) directory of the 1581 Emulation Mode partition, issue this command without the sub-partition name. This will also occur if you issue an Emulation Mode INITIALIZE command to the 1581 Emulation Mode partition.

Please note that if you exit a 1581 Emulation Mode partition with a 'CP' command and then return to it later with another 'CP' command, you will be placed into whichever sub-partition you were in when you exited.

Example of moving to a different 1581 sub-partition:

```
OPEN15,12,15:PRINT#15,"/4:SUB1":CLOSE15
```

JiffyDOS Example:

```
@/4:SUBl
```

Formatting Partitions

The standard Commodore DOS NEW command (not to be confused with the BASIC NEW command) may be used to format partitions on the HD from either BASIC 2.0 or BASIC 7.0. Although the HD will also accept the BASIC 7.0 HEADER command, this command is limited to either the current partition (0) or partition 1.

The DOS NEW commands may be used to delete all files from a partition. It is not necessary to format any partitions on the HD before you begin using them, as they have been pre-formatted at the time of creation. Even when you create new partitions on the system, formatting is performed automatically by HD-TOOLS. You may wish to format them again, however, to change the header name or disk ID. When using the DOS NEW command, the HD can accept both the long and short versions. The

difference in time is slight, especially in comparison to other disk drives (formatting rarely takes longer than a second or two). The BASIC 2.0 or BASIC 7.0 syntax for the DOS NEW command is given below.

```
OPENlf,dv,15:PRINT#lf,"N[n]:partname[,id]"
:CLOSElf
```

where: lf = the logical file number for the command channel
dv = the current device number of the HD
n = the partition you wish to format (0-254)
partname = the name you wish to appear in the partition header
id = a two character id for the partition header

The following syntax applies to the BASIC 7.0 HEADER command:

```
HEADER"partname"[,Iid][,Dn][{ON|,}Udv]
```

where: partname = the name you wish to appear in the partition header
id = a two character id for the partition header
n = the partition number you wish to format (0 or 1)
dv = the device number of the HD

Examples:

```
OPEN15,12,15:PRINT#15,"N3:PARTITION 3,P3":CLOSE15
HEADER"PARTITION 1"IP1,Dl ON U12
```

JiffyDOS examples for using the DOS NEW command:

```
@N3:PARTITION 3,P3
@"N3:PARTITION 3,P3",12
```

Note: The DOS NEW and BASIC 7.0 HEADER commands will not be accepted if issued from within a Native Mode subdirectory.

Formatting 1581 Style Sub-partitions

The DOS NEW and BASIC 7.0 HEADER commands are also used to format the 1581 style sub-partitions which may be created within 1581 Emulation Mode partitions. 1581 sub-partitions must be formatted before they can be used. Before attempting to format a sub-partition, you must make sure that the sub-partition is the current sub-partition within the 1581 Emulation Mode partition that contains it. Do this by using the command outlined in 'Moving Between 1581 Style Sub-partitions' elsewhere in this section. To avoid formatting the wrong area on your drive, it is usually wise to make the appropriate 1581 Emulation Mode partition the current partition.

Initializing Partitions

The DOS INITIALIZE command is often used when a different disk is inserted into a floppy disk drive. This ensures that the drive updates its buffer with a copy of the BAM on the new disk. This function is performed automatically by the HD, but the command has been implemented to retain compatibility. Note that initializing a 1581 Emulation Mode partition causes it to return to the root directory (ala the 1581). The syntax is:

```
OPENlf,dv,15:PRINT#lf,"I[n][:]":CLOSElf
```

where: lf = the logical file number for the command channel
dv = the current device number assigned to the HD
n = the partition to be initialized

Example:

```
OPEN15,12,15:PRINT#15,"13:":CLOSE15
```

JiffyDOS example:

```
@I3:
```

Validating Partitions

The DOS VALIDATE and BASIC 7.0 COLLECT commands check all files in a partition to verify proper allocation of disk space, free any improperly allocated blocks, and delete unclosed (splat '*') files. You should not use these commands on partitions that contain blocks allocated via the BLOCK-ALLOCATE command, or else information may be lost. The VALIDATE command must be used with BASIC 7.0 if you wish to validate a partition other than the current (0) partition or partition 1. The syntax is:

```
OPENlf,dv,15:PRINT#lf,"V[n][:]"CLOSElf
```

where: lf = the logical file number for the command channel
dv = the device number of the HD
n = the partition you wish to validate (0-254)

The syntax for the BASIC 7.0 COLLECT command is:

```
COLLECT[,Dn][{ON|,}Udv]
```

where: n = the partition you wish to validate (0 or 1)
dv = the device number of the HD

Examples:

```
OPEN15,12,15:PRINT#15,"V2:":CLOSE15
COLLECT,DO,U12
```

JiffyDOS example of the DOS VALIDATE command:

```
@V2
```

Partition directory

Having multiple partitions on the hard drive necessitates having the ability to view a directory of partitions. The partition directory may be viewed while you are working within any partition and relates information concerning the number, name, and type of- each partition on the HD. This command also contains options that allow you to specify which partitions will be listed. The syntax for this command is as follows:

```
LOAD"$=P[:*][=tp]",dv
```

where: tp = partition type - N = native
4 = 1541
7 = 1571
8 = 1581
C = 1581 CP/M

dv = the current device number assigned to the HD

Examples:

```
LOAD"$=P",12
LOAD"$=P:*",12
LOAD"$=P:*=8",12
```

JiffyDOS examples:

```
@$=P
@"$=P:*=N",12
```

Renaming Partitions

If you reformat a partition with the DOS NEW command, you may wish to change its name in the partition directory as well. In order to do this we have added the 'Rename Partition' command. This command is similar to the DOS command which is used to rename files. The syntax is as follows:

```
OPENlf,dv,15:PRINT#lf,"R-P:newname=oldname":CLOSElf
```

where: lf = the logical file number
dv = the current device number of the hard drive
newname = the name you wish to assign to the partition
oldname = the name of the partition in the partition directory

Example:

```
OPEN15,12,15,"R-P:WORK=NATIVE 1":CLOSE15
```

JiffyDOS Examples:

```
@R-P:WORK=NATIVE 1
@"R-P:WORK=NATIVE 1",12
```


Renaming Directory Headers

After placing a number of files within a given partition or subdirectory, you may wish to change the name that appears in the directory header (displayed at the top of the directory listing for that partition or subdirectory). Although this could be done by using the DOS NEW command, all files within that partition would be lost at the same time. In order to allow you to rename a header without having to lose or copy all of your files, the HD has a 'Rename Header' command. The syntax is as follows:

```
OPENlf,dv,15:PRINT#lf,"R-H[n][path]:newname":CLOSElf
```

where: lf = the logical file number
dv = the current device number of the hard drive
n = the partition where the header is to be renamed
path = the subdirectory path
newname = the new name for the specified header

Examples:

```
OPEN15,12,15,"R-H:WORK":CLOSE15
OPEN15,12,15,"R-H3:DOWNLOADS":CLOSE15
OPEN15,12,15,"R-H1//ASSEM/:BUDDY64":CLOSE15
```

JiffyDOS Examples:

```
@R-H:WORK
@"R-H//ASSEM/:BUDDY64",12
```

Getting Partition Information

The DOS 'Get Partition Info' command has been created for the purpose of gathering information about the current or some other specific partition. This command will prove valuable to the programmer whose software must react differently to partitions of various types. The partition number for which the information is requested may be placed into a variable and inserted into the command as a character string.The syntax for the G-P command is:

```
OPENlf,dv,15:PRINT#lf,"G-P"[+CHR$(n)]:CLOSElf
```

where: lf = the logical file number for the command channel
dv = the device number of the HD
n = the partition number (0-255)

If the 'G-F' command is sent without the optional character string or a value of 255, the information returned will be for the current partition. A value of 0 requests that the information returned is to be for the system partition. Thirty bytes (0-29) of information concerning the requested partition plus a CHR$(13) are returned over the drive error channel. The following is a list of this information:

Byte 0	-	Partition type	0	-	not created
			1	-	Native Mode
			2	-	1541 Emulation Mode
			3	-	1571 Emulation Mode
			4	-	1581 Emulation Mode
			5	-	1581 CP/M Emulation Mode
			6	-	Print Buffer
			7	-	Foreign Mode
			255	-	System
Byte 1	-	CHR$(0) (reserved)			
Byte 2	-	Partition number			
Byte 3-18	-	Partition name as displayed in the partition directory			
Byte 19	-	Starting system address of partition (high byte)			
Byte 20	-	Starting system address of partition (middle byte)			
Byte 21	-	Starting system address of partition (low byte)			
Byte 22-26	-	CHR$(0) (reserved)			
Byte 27	-	Size of partition (high byte)			
Byte 28	-	Size of partition (middle byte)			
Byte 29	-	Size of partition (low byte)			
Byte 30	-	CHR$(13)			

Note: The values returned in bytes 19-21 and 27-29 are specified in 512 byte blocks. Also keep in mind that any currently undefined bytes may later be used for specific purposes.

Important: To avoid problems with reading information from Partition 13, the G-P command should always be sent with a trailing carriage return (CHR$(13)). The BASIC PRINT# statement will do this for you automatically as long as you do not follow it with a trailing semicolon (;).

Autobooting

It is possible to autoboot from the HD when it is used with a C128 or 128D in 128 mode. To do so, the HD must either be configured as device number 8, or you must issue the BASIC 7.0 BOOT command. You must also make sure that the current partition has a valid boot sector. The boot sector is located at track 1, sector 0 in all partitions, including Native Mode partitions. An interesting benefit in Native Mode partitions is that the boot sector is always allocated. It is therefore never in danger of being overwritten by files, and cannot be freed by the DOS VALIDATE command. The following syntax applies to the BOOT command:

```
BOOT[[Dn][ON|,][Udv]]
```

where:	n	=	the partition where the file is located (0 or 1)
	dv	=	the device number currently in use by the HD

The partition number for this command may only contain a zero or a one. Zero is used to indicate the current partition, while one indicates partition number 1. This command will not accept any other partition numbers, due to a limitation in the BASIC 7.0 command parsing routines. The structure of the boot sector is the same as found on standard Commodore disk drives.

Examples:

```
BOOT U12
BOOT D0,U12
```

Subdirectory Commands

Three new DOS commands have been added to allow you to create and remove subdirectories, as well as to change the current directory. Both the Create and Change commands use a similar syntax, while the syntax for the Remove command has been limited to avoid problems. These commands, like the subdirectories themselves, are very similar to those found in MS-DOS.

Creating Native Mode Subdirectories

The 'Make Directory' command allows you to create Native Mode subdirectories. This command allows you to use standard path syntax. Using a path allows you to create a subdirectory in any native mode partition no matter what your current partition or current directory may be. Here is that syntax:

```
OPENlf,dv,15:PRINT#lf,"MD[n][path]:name":CLOSElf
```

where:	lf	=	the logical file number for the command channel
	dv	=	the current device number assigned to the HD
	n	=	the Native Mode partition where the subdirectory is to be created
	path	=	the path to the subdirectory in which the new subdirectory will be created
	name	=	the name of the new subdirectory

The above syntax may seem slightly confusing, so here are a few guidelines to help you understand how this syntax works:

1. The name of the subdirectory you wish to create must always be separated from the rest of the command by a colon (:).
2. If you are creating a subdirectory within another subdirectory, you must specify that subdirectory within the path unless it is your current directory.

3. If subdirectories are specified within the path of the command (to the left of the colon), each subdirectory name must fall between slash (/) characters (only 1 slash is needed between subdirectory names).
4. Paths normally start at the current directory. If you want the path to start at the root directory (the main directory in that partition), the path should begin with two slashes.
5. If the subdirectory is to be created in a partition other than the partition in which you are located, place the partition number at the start of the path (in front of any slashes).

The following examples should help clarify these guidelines:

```
OPEN15,12,15:PRINT#15,"MD:TEMP":CLOSE15
OPEN15,12,15:PRINT#15,"MD1:TEMP":CL0SE15
OPEN15,12,15:PRINT#15,"MDl//:TEMP":CLOSE15
OPEN15,12,15:PRINT#15,"MD1//TEMP/:TEMP2":CLOSE15
OPEN15,12,15:PRINT#15,"MD/TEMP/:TEMP2":CLOSE15
```

JiffyDOS Examples:

```
@MD:TEMP
@MD1:TEMP
@MD1//:TEMP
@"MDl//TEMP/:TEMP2M"
@"MD/TEMP/:TEMP2",12
```

Moving Between Native Mode Subdirectories

The 'Change Directory' command allows you to move between Native Mode subdirectories. This command employs the same syntax used in the 'Make Directory' command. Using a path allows you to move to a subdirectory anywhere in the currently selected Native Mode partition. This command will also allow you to change the currently selected directory in any other partition, but will not move you into that directory. In order to move to the current directory of a different partition, you must issue a 'Change Partition' command. The 'Change Directory' syntax is:

```
OPENlf,dv,15:PRINT#lf,"CD[n]{[<-]|[[path][:]
subname]}":CLOSElf
```

where:
- lf = the logical file number for the command channel
- dv = the current device number assigned to the HD
- n = the Native Mode partition where the subdirectory you wish to make the current directory exists
- path = the subdirectory path leading to the
- subname = the name of the subdirectory

Note that you can include the back arrow immediately after (CD[n]' to move backwards one directory (to the PARENT). The back arrow cannot be combined with any subdirectory path information. See the examples below.

It is not required that you include the colon before the subdirectory name, as long as the subdirectory name is preceded by a slash.

Here are some examples of the Change Directory command:

```
OPEN15,12,15:PRINT#15,"CD:TEMP":CLOSE15
OPEN15,12,15:PRINT#15,"CD1//:TEMP":CLOSE15
OPEN15,12,15:PRINT#15,"CD1//TEMP/:TEMP2":CLOSE15
OPEN15,12,15:PRINT#15,"CDl<-":CLOSE15
OPEN15,12,15:PRINT#15,"CD/TEMP/TEMP2":CLOSE15
```

JiffyDOS Examples:

```
@"CD:TEMP",12
@CD1//TEMP
@CD1//TEMP/TEMP2
@CDl<-
@CD/TEMP/TEMP2
```

Deleting Native Mode Subdirectories

The 'Remove Directory' command allows you to delete Native Mode subdirectories. This command does not allow the use of paths in order to avoid problems with removing a subdirectory which is a parent of the directory in which you are located. This command will not allow you to delete a subdirectory which contains any files- you must delete these files first by using the DOS SCRATCH or the BASIC 7.0 equivalent The following syntax applies to the 'Remove Directory' command:

```
OPENlf,dv,15:PRINT#lf,"RD[n]:subname":CLOSElf
```

where:
- lf = the logical file number for the command channel
- dv = the current device number assigned to the HD
- n = the partition where the subdirectory you wish to remove exists
- subname = the name of the subdirectory you wish to remove

Here are some examples of the Remove Directory command:

```
OPEN15,12,15:PRINT#15,"RD3:TEMP":CLOSE15
OPEN15,12,15:PRINT#15,"RD:TEMP2":CLOSE15
```

JiffyDOS Example:

```
@"RD3:TEMP", 12
@RD:TEMP2
```

Viewing Directories

Directories may be viewed by using the following BASIC 2.0 command:

```
LOAD "$",12
```

This command will load the current directory from your HD (assuming it is set as device number 12). You may then issue the LIST command to view the directory on your screen. You may also use a partition number for selecting the directory to be loaded, as in the following example:

```
LOAD "$2",12
```

Pattern Matching

Selective directories may also be loaded in the standard way, by placing a colon (:) at the end of the partition number or path, and by using a filename or pattern matching characters to determine which files to include in the listing. The equals (-) sign and a filetype designator may also be included after the filename to indicate a particular filetype. The filetype characters are: P for program (PRG), S for sequential (SEQ), U for user (USR), R for relative (REL), and B for subdirectory branch (DIR).

Examples:

```
LOAD "$2:S*=P",12
```

This example will load a directory of all PRG files (=P) starting with an S (S*) from partition number 2. You may also use the asterisk at the beginning of a filename as in the following example:

```
LOAD "$1/UTILS/:*E=P",12
```

This example will load a directory of all PRG files which end with an 'E'. The question mark '?" may also be used to replace an unknown character in the filename. It is also possible to use the asterisk in the middle of a pattern as shown in this example:

```
LOAD "$1/UTILS/:R*E=P",12
```

This pattern will match filenames like RIDE and RUE. Only one asterisk may be used in the pattern. Another matching character is the question mark which will match any character found at that position in the filename.

```
LOAD "$2:B?RE=P",12
```

This example will load a directory of all PRG files which are four characters long, start with 'B' and end with 'RE'. More than one question mark may be used in a pattern. It is also possible to mix question marks and an asterisk together in a pattern.

Time and Date Stamped Directory Listings

Since the HD contains real time clock circuitry, it was possible to easily place each file's time and date of creation into the directory entry. This is done for all file types, regardless of the partition type in which it was created. The HD uses the same method as GEOS for time and date stamping file entries. In order to allow the user to easily view the time and date stamp, new options have been added for loading directories. These options allow the stamp to be viewed, and also permit the user to select files which were created within a specified timeframe. The syntax for the time and date stamped directory is:

```
LOAD"$=T[n][path][:pattern[={tp|option}[,option]]],dv
```

where:

	n	=	partition number of the directory to be loaded
	path	=	the subdirectory you wish to view
	pattern	=	name of file or pattern to match
	tp	=	first character of filetype (P, S, R, U or B)
	option	=	one of the options listed below
	dv	=	current device number assigned to the HD
options:	L	=	long format
	N	=	do not include time and date in listing
	>*stamp*	=	greater than or equal to *stamp*
	<*stamp*	=	less than or equal to *stamp*

stamp format: *MM/DD/YY HH:MM x*M

Although the syntax for this command may look a little complex, it is really quite simple to use when broken down into separate elements.

The partition number (*n*) may be specified if desired. If this parameter is omitted, the current partition will be targeted for this command.

The *filename* is the name of a file or a standard pattern matching string. This means you may use the asterisk (*) to match a number of characters, and question marks (?) to match individual characters. See 'Pattern Matching' under the subsection 'Viewing Directories' for examples.

The file type (*tp*) is optional, but if specified it must be the first option after the filename pattern. This may be a P (PRG), S (SEQ), R (REL), or U (USR). If you wish to view all file types, skip this option. This too is covered in the examples under the 'Pattern Matching' heading.

The options allow you to match specific times and dates (>*stamp*,<*stamp*), and also to specify long formal (L). You may also specify that the directory entries match a certain time and date stamp, but that the directory list is not to include these limes and dates (N). You may use as many of these options as you wish, but they must be separated by commas (,).

The 'long' time format gives the full date and time:

```
112 "TESTFILE"          PRG     02/02/90   04.44 PM
```

The 'short' (default) time format gives the date and time as follows:

```
112 "TESTFILE"          P 02/02   04.44 P
```

If the no-list (N) option is given, the directory entries will be loaded as they normally appear whether the long format (L) is specified or not This means that specifying the long format and the no-list option in the same command is usually a waste of time. The reason that the no-list option was created was to allow you to use the time and date of files as pattern matching criteria within programs which cannot accept the. extra time and date characters.

The *<stamp* option will list all files which have a creation time and date less than or equal to the time and date specified in *stamp*. The *>stamp* option will list all files which have a creation time and date greater than or equal to the time and date specified in *stamp*.

If both the *<stamp* and *>stamp* options are used within the same command, the resulting list of files will include files which fall between the range of the two time and date stamps specified.

The *stamp* format must be entered exactly as shown. This means you must specify the month, day, and year with two characters each and separate them with a slash (/). The hour and minute must also be given with two characters each in 12 hour format separated with a colon (:) or a period (.). The last parameter must be AM or PM. The date and time must be separated by a single space, and so must the time and AM/PM parameters. Here are a few examples:

```
LOAD"$=T",12
LOAD"$=T2",12
LOAD"$=T2:*=P", 12
LOAD"$=T2:*=P,L",12
LOAD"$=T2:*=P,L,>12/21/89 04:15 PM",12
LOAD"$=T:*=L,<12/21/89 04:15 PM",12
LOAD"$=T4:*=S,N,>12/01/89 12:01 AM,<12/31/89
12:00 PM",12
```

JiffyDOS Examples:

```
@"$=T",12
@"$=T2"
@$=T2:*=P
@$=T4:*=S,N,>12/01/89 12:01 AM,<12/31/89 12:00 PM
```


File Commands

File commands are the most commonly used commands. They include loading and saving files, verifying, renaming, scratching, copying, and locking files and are entered in much same manner as for other Commodore or compatible disk drives. Although the BASIC 2.0 versions of these commands are supported in BASIC 7.0, BASIC 7.0 also contains other commands that perform the same functions. Note that the BASIC 2.0 versions of these commands allow more versatility when dealing with partitions.

Loading Files

The following syntax can be used to load programs in BASIC 2.0 and BASIC 7.0:

```
LOAD"[[n][path]:]filename",dv[,sa]
```

where:	n	=	is any legal partition number from 1 to 254
	path	=	the subdirectory path to the file
	filename	=	is any legal filename of up to 16 characters
	dv	=	is the current device number assigned to the HD
	sa	=	is the secondary address if needed

To load a machine language program, a secondary address of 1 must be added to the end of this command, separated from the device number by a comma.

Examples:

```
LOAD "2:BASEBALL",12
LOAD "3/TERMS/:TERMBOOT",12,1
```

JiffyDOS examples:

```
/"2:BASEBALL", 12
/2:BASEBALL
%3/TERMS/:TERMBOOT
```

It is generally a good idea to use the BASIC 2.0 syntax if you are specifying partitions since BASIC 7.0 will only allow access to the current (0) partition or partition 1, and will not allow the use of subdirectory paths.

The BASIC7.0 BLOAD command can be used to load machine language or data files into memory. The DLOAD command is used primarily to load BASIC programs. The syntax for these commands is shown below.

```
BLOAD"filename"[,Dn][{ON|,}Udv][,Bb][,Pa]
DLOAD"filename"[,Dn][{ON|,}Udv]
```

where: filename = the name of the machine language program to be loaded
n = the partition number where the file is located (0 or 1)
dv = the device number currently in use by the HD
b = the memory bank where the file is to be loaded
a = the starting address of the file to be loaded

Examples:

```
BLOAD"SPRITE",D0,U9,B0,P3584
DLOAD"TEST"/D0 ON U9
DLOAD"TEST2"
```

Saving Files

The following syntax can be used to save programs in BASIC 2.0 and BASIC 7.0:

```
SAVE "[[@][n][path]:]filename",dv
```

where: n = is any legal partition number from 1 to 254
path = the subdirectory path where the file is to be saved
filename = is any legal filename of up to 16 characters
dv = is the current device number assigned to the HD

The '@' symbol shown in the command syntax may be used to indicate that a file with the same name which already exists should be replaced with the new file. This is called the 'Save with Replace option and if it is used, it must be followed by a partition number and a colon (:). To save a machine language program from a C64 or C128 in 64 mode, you must use a machine language monitor or change some of the BASIC pointers.

Examples:

```
SAVE"2:BASEBALL",12
SAVE"/TERMS/:TERMBOOT",12
```

JiffyDOS examples:

```
<-"2:BASEBALL",12
<-/TERMS/:TERMBOOT
```


You may use the BASIC 7.0 BSAVE and DSAVE commands when it is your intention to work with your current partition (0) and directory, or in the current directory of partition 1. BSAVE is intended for flies other than BASIC programs, while DSAVE is intended for BASIC programs.

```
BSAVE"[@]filename"[,Dn][{ON|,}Udv][,Bb],Pa TO Pe
DSAVE"[@]filename"[,Dn][{ON|,}Udv]
```

where: filename = the name of the file to be saved
n = the partition number where the file is to be saved
dv = the device number currently in use by the HD
b = the memory bank where the file is to be saved
a = the starting address of the file to be saved
e = the ending address of the file to be saved

The '@' symbol may be included to indicate that the file being saved is to replace an existing file with the same. This is called 'Save with Replace'.

Examples:

```
BSAVE"SPRITE",DO,U9,BO,P3584
DSAVE"TEST",D1 ON U9
DSAVE"TEST2"
```

Verifying Files

BASIC 2.0 and 7.0 contain commands which allow you to verify if a program has been saved properly. These commands compare the saved program with the contents of memory. Keep in mind that any change in the contents of memory may cause a verify operation to fail. It is best to verify a file immediately after saving it for this reason. Both versions of BASIC support specifying a partition within the filename portion of this command (as described earlier). The following syntax applies to these commands:

```
VERIFY"[[n][path]:]filename",dv[,sa]
```

where: n = the partition where the file is located
path = the subdirectory path to the file you wish to verify
filename = the name of the file to be verified
dv = the current device number assigned to the HD
sa = secondary address of 1 to verify a non-BASIC file

Examples:

```
VERIFY"NEWSTATS",12
VERIFY"1/UTILS/TERMS/:XLATOR",12
```

JiffyDOS example:

```
'"NEWSTATS",12
```

In BASIC 7.0, the standard VERIFY command is accepted, but you may also use the DVERIFY command. This command is limited to use with the current partition (0) or partition 1, and the current subdirectory.

```
DVERIFY"filename"[,Dn][{ON|,}Udv]
```

where: filename = the name of the file to be verified
n = the partition where the file resides (0 or 1)
dv = the device number of the HD

Example:

```
DVERIFY"NEWSTATS",D1,U12
```

Renaming Files and Subdirectories

Filenames and Native Mode subdirectory names may be changed by using either the DOS RENAME or the BASIC 7.0 RENAME command. The BASIC 7.0 version only supports the current directory within the current partition (0) or partition 1. When using either version, the partitions specified for the two file names must either be the same, or must indicate the same partition. The syntax for the DOS RENAME command is:

```
OPENlf,dv,15:PRINT#lf,"R[n][path]:newname=[[n]
[path]:]filename":CLOSElf
```

where: lf = the logical file number for the command channel
dv = the current device number of the hard drive
n = the partition where the file to be renamed is located
path = the subdirectory path to the file you want to rename
newname = the new name to be assigned to the file
filename = the name of the file which is being renamed

Examples:

```
OPEN15,12,15:PRINT#15,"R1:BOOT1=BOOT":CLOSE15
OPEN15,12,15,"R1/UTILS/:NEWT=1/UTILS/:WW":CLOSE15
```

JiffyDOS Example:

```
@"R1:BOOT1=BOOT", 12
```

The BASIC 7.0 RENAME command syntax is:

```
RENAME[Dn,]"filename"TO[Dn]"newfile"[,Udv]
```

where: n = the partition where *filename* is located
filename = the name of the file which is being renamed
newfile = the new name to be assigned to the file
dv = the current device number of the hard drive

Scratching (deleting) Files

The standard DOS and BASIC 7.0 SCRATCH commands may be used to delete files from a partition. As with many of the other BASIC 7.0 commands, SCRATCH is only effective with partition numbers 0 (current) and 1. The standard Commodore DOS SCRATCH may be used in place of the BASIC 7.0 version when necessary and with BASIC 2.0. The following command syntax covers the DOS version of this command:

```
OPENlf,dv,15:PRINT#lf,"S[n][path]:filename[,[n]
[path]:]filename...]":CLOSElf
```

where: lf = the logical file number for the command channel
dv = the current device number of the hard drive
n = the partition(s) which hold the file(s) to be scratched
path = the subdirectory path(s) to the file(s)
filename = the name of the file(s)

Multiple files may be scratched with this command which will accept up to five separate filename parameters. Different partitions can be specified with the separate filenames. The filename parameters may also contain wildcards to allow scratching of multiple files within a single partition.

Examples:

```
OPEN15,12,15:PRINT#15,"S1:JUNK,3:C?*.BAS":CLOSE15
OPEN15,12,15,"S1/UTILS/: CO*":CLOSE15
```

JiffyDOS Examples:

```
@"S1:JUNK,3:C?*.BAS",12
@S1/UTILS/:CO*
```

The BASIC 7.0 SCRATCH command syntax is:

```
SCRATCH"filename"[,Dn[{On|,}Udv]
```

where: filename = the name of the file to be scratched
n = the partition where the file to be scratched resides
dv = the device number of the HD

Multiple files may also be scratched with this command by using pattern matching, although it does not allow you to specify multiple file names as does the DOS version. Remember, this command is only valid for use with the current directory in partition numbers 0 (current) or 1.

Copying Files

Copying is an important consideration with any storage device. For this reason the HD comes supplied with DOS commands which allows you to copy files between partitions. This function may also be accomplished by using the FCOPY program supplied with the HD. Another copy program supplied with the HD (MCOPY) allows you to copy an entire disk to a similar partition on the HD and vice versa. For more information on the copy programs included with the HD, see Appendix A.

Copying files between drives

Files may be copied between the HD and other disk drives using a standard file copier. Only generic copiers that do not try to discover the drive type by checking ROM locations will work with the HD.

We have included FCOPY with the HD to assist in file copying. FCOPY will work with all file types and all drive types including an REU running under RAMDOS.

you have JiffyDOS installed on your computer, you may use the built-in Jiffy-DOS file-copier with the HD as well. Many of the commercial copy programs will not work with the HD because they look at specific memory locations to try to identify the drive being used, or attempt to write drive specific code into the disk drive to speed up the copy process.

Copying and Combining files between partitions

You may copy files from one partition to another on the HD by using the standard Commodore DOS COPY command. This command allows you to place a partition number in front of each of the filenames specified in the command. The syntax for this command is as follows:

```
OPENlf,dv,15:PRINT#lf,"C[n][path]:newfile=[[n]
[path]:]filename[,[n:]filename...]":CLOSElf
```

where:
- lf = the logical file number for the command channel
- dv = the current device number of the hard drive
- n = the partition which holds or is to receive the file
- path = the subdirectory path(s) where the file(s) to be copied or created is (are) located
- newfile = The name of the new file being created
- filename= the name of the file(s) which is (are) being copied

Up to five files may be combined into a single file by using this command, though it is important to note that copying a number of files into a single file is only effective with text files. If you use this command for copying a single file from one file to another file in a different partition, you may use the same filename for both files.

Examples:

```
OPEN15,12,15:PRINT#15,"C1:FCOPY=3:FCOPY":CLOSE15
OPEN15,12,15,"C:FULLSTATS=STAT1,3:STAT3":CLOSE15
OPEN15,12,15,"C2:MCOPY=1/COPIERS/:MCOPY":CLOSE15
```

JiffyDOS Examples:

```
@"C1:FC0PY=3:FCOPY",12
@"C2:MCOPY=1/COPIERS/:MCOPY"
@C:FULLSTATS=STATS1,3:STATS3
```

You may also copy files from one partition to another on the HD by using the standard BASIC 7.0 COPY command. This command is limited to copying files in the current directory of the current partition (0) or partition 1. Use the DOS COPY command mentioned earlier if you want to copy files between other partitions. The syntax for this command is as follows:

```
COPY[Dn,]"filename"TO[Dn]"newfile"[,Udv]
```

where:
- n = the partition which holds or is to receive the file
- filename = the name of the file(s) which is being copied
- newfile = the name of the file being created
- dv = the current device number of the hard drive

If you use this command to copy a file to a different partition, you may use the same filename for both flies. If they are to reside in the same partition, you must use different filenames or an error will result

Two files may be combined into a single file by using the BASIC 7.0 CONCAT command, although it is important to note that adding flies together in this manner is only effective with text files. The BASIC 7.0 limitation of using only the current directory of the current partition (0:) or partition 1 applies to this command. The syntax is:

```
CONCAT[Dn,]"filename"TO[Dn]"newfile"[,Udv]
```

where:
- n = the partition where the file exists
- filename = the name of the file which is being added
- newfile = the name of the file being added to
- dv = the current device number of the hard drive

The file previously named *newfile* will be replaced by the newly created combined file. The only way these files may have the same name is if they exist in different partitions.

Examples:

```
CONCAT "NEWNUMBERS" TO "ALLNUMBERS"
```

Locking and Unlocking Files

Files located on the HD may be locked to avoid scratching them by accident Before you can scratch a locked file, it must first be unlocked. Locking a file sets one of the bits in the filetype byte for that file (located in the directory entry of that file). Files which have been locked will appear in the directory with a 'less-than' symbol to the right of the filetype. For example, a file named JIFFYMON which has been locked will appear in the directory listing as:

```
33   "JIFFYMON"          PRG<
```

It is also possible to lock Native Mode subdirectories and 1581 sub partitions. If a subdirectory has been locked, it is not possible to delete it with the 'Remove Directory' command until it has been unlocked.

The Lock command is a 'toggle' function. Using it on an unlocked file will cause the file to become locked. Using it on a file which has already been locked will unlock that file. The syntax for locking and unlocking files is:

```
OPENlf,dv,15:PRINT#lf,"L[n][path]:name":CLOSElf
```

where:	lf	=	the logical file number
	dv	=	the current device number of the hard drive
	n	=	the partition number in which the file exists
	path	=	the Native Mode subdirectory path in which the file exists
	name	=	the name of the file or subdirectory you wish to lock or unlock

The following examples illustrate the use of this command:

```
OPEN15,12,15:PRINT#15,"L:TEST":CLOSE15
OPEN15,12,15:PRINT#15,"L1//:TEST":CLOSE15
OPEN15,12,15:PRINT#15,"L/UTILS/:TEST":CLOSE15
```

JiffyDOS examples:

```
@L:TEST
@"L1//:TEST"
@"L/UTILS/:TEST",12
```

Note: JiffyDOS contains its own version of the LOCK command. This may also be used with the HD. See your JiffyDOS manual for details on using this version.

Relative File Commands

Relative files are files which contain an index table to allow quicker access to a particular portion of the file called a record. Records are kept track of by a special section of the relative file called a side sector. Two different types of side sectors exist in the HD: Regular side sectors (the default type used in 1541 and 1571 Emulation Mode partitions), and super side sectors (the default in 1581 Emulation Mode and Native Mode partitions).

Relative files may be up to 720 blocks long when regular side sectors are used. Relative files which use a super side sector may be over 60,000 blocks long. Up to 65,535 records are allowed and each record may be from 2 to 254 characters in length. All records in the same file are the same length, although it is not necessary to use all of the characters in each record.

The main advantage of using a relative file is speed. Normally, you must read from the beginning of a file until the information you wish to find is reached. With relative files, as long as you keep track of which record your information is stored in, you may go directly to that area of the file and read or write the required information right away. A separate index is usually kept in another file type to keep track of the particular information that each record contains.

Opening or Creating a Relative File

The same syntax can be used to create new or open existing relative files by using the BASIC 2.0 OPEN command or the BASIC 7.0 DOPEN command. When creating a new relative file, print a new record to the file after opening it, using the PRINTS command. When you are done accessing the relative file, close it with the CLOSE command, or the DCLOSE command if you used the BASIC 7.0 syntax to open it. Here is the BASIC 2.0 syntax for opening or creating a new relative Hie:

```
OPENlf,dv,sa,"[[n][path]:]filename[{"|,L"
+CHR$(rl))]
```

where:	lf	=	the logical file number
	dv	=	the current device number of the hard drive
	sa	=	the secondary address (2 through 14)
	n	=	the partition in which the file exists or is to be created
	path	=	the Native Mode subdirectory path which leads to the file
	filename	=	the name of the relative file
	rl	=	the record length (only needed when creating a new relative file)

Examples:

```
OPEN2,12,2,"1/DATA/:CUSTOMERS,L"+CHR$(127)
OPEN2,12,2,"ADDRESS"
```

This syntax may also be used in BASIC 7.0, but an alternate syntax was provided for this newer version of BASIC. Again, as with all BASIC 7.0 specific commands, you will be limited to using the current partition (0) or partition 1, and you will not be able to specify subdirectories. Here is that alternate syntax:

```
DOPEN#lf,"filename"[,Lrl][,Dn][,Udv]
```

where:
- lf = the logical file number
- filename = the name of the relative file
- rl = the record length (only needed when creating a new relative file)
- n = the partition in which the file exists or is to be created (only 0 or 1 is accepted)
- dv = The current device number of the hard drive

Examples:

```
DOPEN#2,"CUSTOMERS",L127,D1,U12
DOPEN#2,"ADDRESS"
```

Positioning to a Specific Record

When you are ready to read or write a specific record, it is necessary to use either the DOS POSITION command, or the BASIC 7.0 RECORD command. The relative file must already be opened in order to use these commands, and in the case of the DOS POSITION command, you must also have the command channel open to the drive. These commands may also be used to create a number of blank records when first creating a relative file. This makes writing the actual data much faster. Here is the syntax for the DOS POSITION command:

```
PRINT#lf,"P"+CHR$(ch)+CHR$(lr)+CHR$(lh)[+CHR$(of)]
```

where:
- lf = the logical file number for the command channel
- ch = the secondary address used when opening the relative file plus a value of 96
- lr = the low byte of the record number you wish to access or create
- lh = the high byte of the record number you wish to access or create
- of = the byte number in the record which you wish to start reading from or writing to (first byte if left out)

Examples:

```
PRINT#15,"P"+CHR$(98)+CHR$(30)+CHR$(0)+CHR$(10)
PRINT#15,"P"+CHR$(98)+CHR$(30)+CHR$(0)
```

Here is the syntax for the BASIC 7.0 RECORD command:

```
RECORD#lf,rn[,of]
```

where:
- lf = the logical file number for the relative file
- rn = the record number you wish to access or create
- of = the byte number in the record which you wish to start reading from or writing to (first byte if left out)

Examples:

```
RECORD#2,30,10
RECORD#2,30
```

If you are creating a relative file ahead of time, use the PRINT# command to print a CHR$(255) to the last record. This character has a special meaning in relative files- it is used as the first character for empty records. You may expand the size of the relative file later if you run out of records simply by writing a record with a higher number than currently exists in the file. It just lakes a little longer to write a record which has not been previously created.

Whenever a new record is created, an error will occur in the drive. This is error number 50 which means "RECORD NOT PRESENT". Obviously, if it was your intent to create this record, this error can be ignored.

Note: Although it is not necessary to send the RECORD or POSITION commands twice on the HD to avoid data corruption, this practice should be followed anyway to avoid problems when using other drives with your program. To do this, send the RECORD or POSITION command once before writing a record as you normally would, and once again afterward. This will help to ensure that your data will not be corrupted due to the flaw which exists in other disk drives.

Special HD Commands

A number of commands have been provided in HD DOS to allow you to take advantage of special features built into this unique system.

Software SWAP Commands

The HD allows you to perform the SWAP 8 and SWAP 9 functions from within software in addition to using the front panel switches. You may also undo any currently active SWAP condition with a SWAP TO DEFAULT command. These commands have been included to allow loaders or boot programs to easily swap the HD's device number. An example of such a loader can be seen by loading and listing BOOTQ - a program intended to allow you to load and execute the Q-Link software from partition number 6.

Whenever one of the SWAP commands is used within a program, it is very important to include a delay loop after issuing the command, even before you attempt to close the command channel. Since these commands instruct the HD to send commands to another drive on the system via the serial bus, any attempt by the computer to use the serial bus while this is taking place could cause a bus collision. Although we have found that delay values of 60 or higher within a FOR/NEXT loop will work with most systems, it is recommended that you use a larger value (500 would be good) to leave ample time for these commands to complete their tasks.

Note that the SWAP commands will not operate under certain conditions, usually when there is a file open to a serial bus device In this case, it would be unwise to perform a SWAP anyway. To avoid these kinds of problems, make sure that you send the SWAP command only when no other files (with the exception of the command channel) are open.

Sending a SWAP TO DEFAULT command is not required when you wish to change from a SWAP 8 condition to a SWAP 9 condition, or vice-versa (the HD will swap directly from 8 to 9 or from 9 to 8). SWAP TO DEFAULT is only needed when you wish to return the HD to it's default device number. It is also possible to send the swap commands from direct mode. You may also abbreviate the command if you are using JiffyDOS. The following syntax applies to the SWAP commands:

```
FORt=1TO500:NEXT:CLOSElf
OPENlf,dv,15:PRINT#lf,"S-x":
```

where:
- lf = the logical file number
- dv = the current device number of the hard drive
- x = 8 to swap with device number 8
9 to swap with device number 9
D to return the HD to its original device number
- t = A variable to be used for the timing loop

Here is a sample program using all of the SWAP commands. This program assumes that the HD is device number 12 at the time the program begins.

```
100 OPEN15,12,15
110 PRINT#15,"S-8":FORI=1TO500:NEXT:CLOSE15
120 OPEN15,8,15
130 PRINT#15,"S-9":FORI=1TO500:NEXT:CLOSE15
140 OPEN15,9,15
150 PRINT#15,"S-D":FORI=1TO5OO:NEXT:CLOSE15
```

Direct mode example:

```
OPEN15,12,15,"S-8"
CLOSE15
```

Note: Using these commands in direct mode will not require the timing loop as long as you send the SWAP command first, and then close the channel on a second line. By the time you finish typing the CLOSE command, the SWAP function will be done using the serial bus.

JiffyDOS examples:

```
@S-8
@"S-9",8
```

Software Write Protect Commands

The HD allows you to set and release the WRITE PROTECT function via software. It is important to note that, for data security reasons, a write protect condition may not be by-passed via the software command if it has been activated manually from the front panel. For more information on the use of the WRITE PROTECT function, see the heading 'Write Protecting the Drive' in the 'Special Features' section of the manual. The syntax for enabling and disabling the WRITE PROTECT function via software is:

```
OPENlf,dv,15:PRINT#lf,"W-x":CLOSElf
```

where:
- lf = the logical file number
- dv = the current device number of the hard drive
- x = 1 to enable WRITE PROTECT, 0 to disable

Examples:

```
OPEN15,12,15,"W-1":CLOSE15
OPEN15,12,15,"W-0":CLOSE15
```

JiffyDOS examples:

```
@W-1
@"W-1",12
```

Real Time Clock Commands

There are three types of commands provided for reading and setting the HD's internal real-time clock. Each type of command uses a different format for sending and receiving clock data. The data types used are ASCII, BCD (binary coded decimal), and decimal

Reading Time and Date in ASCII Format

The 'T-RA' command allows you to read the HD's clock and return the date and time as an ASCII string over the error channel. The syntax for this command is as follows:

```
OPENlf,dv,15:PRINT#lf,"T-RA"
```

where: lf = the logical file number
dv = the device number of the HD

After the 'T-RA' command is sent, the HD error channel will return the date and time in the following format:

```
"dow. mo/da/yr hr:mi:se xM"+CHR$(13)
```

where: dow = the day of the week (4 characters followed by a space).
SUN.
MON.
TUES
WED.
THUR
FRI.
SAT.
mo = the month (01-12)
da = the day
yr = the year
hr = the hour (01-12)
mi = the minute (00-59)
se = the second (00-59)
x = A or P (denoting AM or PM)

To read the error channel from BASIC, the following GET loop can be used:

```
10 GET#lf,A$:T$=T$+A$:IF ST<>64 THEN 10
```


Writing Time and Date in ASCII Format

The 'T-WA' command allows you to set the HD's internal real-time clock by sending an ASCII string representing the current time over the command channel. The syntax for this command is as follows:

```
OPENlf,dv,15
PRINT#lf,"T-WAdow. mo/da/yr hr:mi:se xM"
CLOSElf
```

where: lf = the logical file number
dv = the device number of the HD

The remaining parameters (*dow.*, *mo*, *da*, etc.) follow the same format as described above under the 'T-RA' command. Note: it is very important that the time and date parameters are separated by the same number of spaces and delimiters as shown above. Also, the day of week *must* be four characters long and followed by a space (see 'T-RA' for valid day-of-week strings). If these parameters are not provided in the correct manner, the HD will not set the time correctly.

Reading Time and Date in Decimal Format

The T-RD command allows you to read the HD's clock and return the date and time as a series of decimal-valued bytes over the error channel. This command provides BASIC (or ML) programmers with a means to read the current time and date in numeric format from within a program. The syntax for the 'T-RD' command is as follows:

```
OPENlf,dv,15:PRINT#lf,"T-RD"
```

where: lf = the logical file number
dv = the device number of the HD

After the 'T-RD' command is sent, the HD error channel will return the date and time as bytes in the following format:

Byte 0 - day of week (00=SUN..01=MON.,etc.)
Byte 1 - year (i.e 1990=90)
Byte 2 - month (01-12)
Byte 3 - day (01-xx)
Byte 4 - hour (01-12)
Byte 5 - minute (00-59)
Byte 6 - second (00-59)
Byte 7 - AM/PM flag (00=AM, non-0=PM)
Byte 8 - CHR$(13)

Writing Time and Date in Decimal Format

The 'T-WD' command allows you to set the HD's internal real-time clock by sending a series of decimal-valued bytes representing the current time over the command channel. The syntax for this command is as follows:

```
OPENlf,dv,15
PRINT#lf,"T-WD"+CHR$(byte0)+CHR$(byte1)+
CHR$(byte2)+CHR$(byte3)+CHR$(byte4)+CHR$(byte5)+
CHR$(byte6)+CHR$(byte7):CLOSElf
```

where: lf = the logical file number
dv = the device number of the HD
bytes 0-7 = the current time and date represented by eight decimal bytes (format given under the 'T-RD' command).

Reading Time and Date in BCD Format

The 'T-RB' command allows you to read the HD's clock and return the date and time as a series of BCD bytes over the error channel. The syntax is:

```
OPENlf,dv,15:PRINT#lf,"T-RB"
```

where: lf = the logical file number
dv = the device number of the HD

The error channel will return the date and time as BCD bytes in the following format:

Byte 0 - day of week(00=SUN.,01=MON.,etc.)
Byte 1 - year (i.e1990=$90)
Byte 2 - month ($01-$12)
Byte 3 - day ($01-xx)
Byte 4 - hour ($01-$12)
Byte 5 - minute ($00-$59)
Byte 6 - second ($00-$59)
Byte 7 - AM/PM flag (00=AM, non-00=PM)
Byte 8 - $0D

Writing Time and Date in BCD Format

The 'T-WB' command allows you to set the HD's internal real-time clock by sending a series of BCD bytes representing the current time over the command channel. This command is normally sent from within a machine language program. The syntax for this command is as follows:

```
"T-WB"+BCD time
```

where: BCD time = The current time and date represented by eight BCD bytes, (format given under 'T-RB').

Sending SCSI Commands

The HD has a built-in command which will allow those familiar with the SCSI Common Command Set to send these commands directly to the HD. We will not attempt to discuss all the various commands which make up the SCSI command set- doing so could easily double the size of this manual, and very few users would be served by this information. We will describe here the DOS command necessary to send SCSI commands to the host controller located in the HD. We call this the 'SCSI Send' command, and the syntax for this command is:

```
OPENlf,dv,15:PRINT#lf,"S-C"CHR$(de)CHR$(bl)
CHR$(bh)CHR$(cb)...
```

where: lf = the logical file number for the command channel
dv = the device number of the HD
de = the device number of the SCSI device
bl = the low byte of the SCSI data buffer in HD RAM
bh = the high byte of the SCSI data buffer in HD RAM
cb = the SCSI command bytes

Reading the error channel immediately after sending a SCSI command to the HD will return a single byte representing the SCSI command completion status. Status byte values and definitions are given below:

00 = OK
02 = check condition
08 = target busy
16 = intermediate status OK
24 = reservation conflict
48 = DOS syntax error

Direct Access Commands

Most direct access commands require that files be opened to both the command channel and to a direct access file. Before using the commands described in this section, you should be familiar with the methods required to open and access the command channel and a file data channel.

The Direct Access Channel

Opening a direct access channel requires the following BASIC statement

```
OPENlf,dv,sa,"#[bu]"
```

where:	lf	=	the logical file number
	dv	=	the device number of the HD
	sa	=	the secondary address
	bu	=	the drive buffer to be used

The logical file number can be any number from 1 to 127, but cannot be the same any logical file currently in use.

The device number is the one currently assigned to the HD (8-29).

The secondary address (also referred to as the channel number) may be any number from 2 to 14. The channel number should be different than any other currently active channel number on the HD. It is usually a good practice to use the same number for the logical file number and the channel number. This usually makes keeping track of files easier for the programmer.

The drive buffer number may currently be any number from 0 through 29. This is an optional parameter, and if left out, the drive will automatically assign the next available buffer. It is usually best to leave the buffer number out unless you need to use a specific one. If you select the buffer number yourself, be sure to check for an error, since selecting a buffer which is already in use will produce an error condition. This example show how to open a direct access file:

```
OPEN2,12,2,"#"
```

> Note: Direct access files are always opened to the current partition number on the HD. If you wish to open a direct access file to a partition other than the one you are currently in, you must change partitions with the 'CP' command first. All further access to this file will occur in that partition number, even if you change partitions after opening the file.

Reading and Writing Data with Direct Access

When BLOCK-READ or BLOCK-WRITE commands are being used, information is normally read or written using the BASIC commands GET#, INPUT#, and PRINT#. The data being accessed with these commands is not being written to or read from the disk directly, but is actually stored in a buffer which temporarily holds the data for a particular block on disk. This may seem a little confusing to those who have never used these commands before, so here is a simpler way of looking at the process.

Reading Data from the Drive

If you wish to read data directly from the disk, a BLOCK-READ or Ul command is sent to the drive controller to tell it to read a particular block from the disk. This block is placed into a 256-byte RAM buffer in the disk drive. To transfer this data from the drive to the computer, the BASIC commands GET# or INPUT# are used. If you only need a portion of the data in the block, you can use the BUFFER-POINTER command to tell the drive which byte will be the first to be passed to the computer with GET# or INPUT#.

Here is a quick example of reading the seventh byte from track 1 sector 0 of device number 12. The buffer points to byte 6 because the byte numbers start at 0, so byte 6 is actually the seventh byte.

```
10 0PEN15,12,15:REM OPEN COMMAND CHANNEL
20 OPEN2,12,2,"#":REM DIRECT ACCESS CHANNEL
30 PRINT#15,"U1";2;O;1;O:REM TRACK 1 SECTOR 0
40 PRINT#15,"B-P";2;6:REM SEVENTH BYTE (6)
50 GET#2,A$:REM READ THE BYTE INTO A$
60 CLOSE2:REM CLOSE DIRECT ACCESS CHANNEL
70 CLOSE15:REM CLOSE COMMAND CHANNEL
```

If you know which buffer is being used, a MEMORY-READ command could also be used for this purpose. Here is an example of this method:

```
10 OPEN15,12,15:REM OPEN COMMAND CHANNEL
20 OPEN2,12,2,"#2":REM USE BUFFER #2
30 PRINT#15,"U1";2;O;1;O:REM TRACK 1 SECTOR 0
40 PRINT#15,"M-R"CHR$(6)CHR$(5) :REM LOC $0506
50 GET#15,A$:REM READ THE BYTE INTO AS
60 CLOSE2:REM CLOSE DIRECT ACCESS CHANNEL
70 CLOSE15:REM CLOSE COMMAND CHANNEL
```

Writing Data to the Drive

If you wish to write data directly to the disk, the PRINT# command is first used to write this data to a 256-byte buffer in the drive. After the data has been written to this buffer, the buffer itself is then transferred to the desired track and sector location on the disk by using the BLOCK-WRITE or U2 command.

Here is a quick example of writing to the seventh byte on track 1 sector 0 of device number 12. As in the previous example, the buffer points to byte 6 because the byte numbers start at 0, so byte 6 is actually the seventh byte.

```
10 0PEN15,12,15:REM OPEN COMMAND CHANNEL
20 OPEN2,12,2,"#":REM DIRECT ACCESS CHANNEL
30 PRINT#15,"B-P";2;6:REM SEVENTH BYTE (6)
40 PRINT#2,CHR$(65):REM WRITE BYTE TO BUFFER
50 PRINT#15,"U2";2;0;1;0:REM TRACK 1 SECTOR 0
60 CLOSE2:REM CLOSE DIRECT ACCESS CHANNEL
70 CLOSE15:REM CLOSE COMMAND CHANNEL
```

The above example writes a sector to the disk with no regard for what existed in that sector previously. If you want to change a byte in the sector without changing the rest of the contents of that sector, read it into the buffer first, change the desired byte to the new value, and then write the sector back to disk.

If you know which buffer is being used to write the data, a MEMORY-WRITE command might also be used for this purpose. Here is an example of this method:

```
10 OPEN15,12,15:REM OPEN COMMAND CHANNEL
20 OPEN2,12,2,"#2":REM USE BUFFER #2
30 PRINT#15,"M-W"CHR$(6)CHR$(5)CHR$(1)CHR$(65)
:REM LOC $0506, ONE BYTE, VALUE 65
40 PRINT#15,"U2";2;0;1;0:REM TRACK 1 SECTOR 0
50 CLOSE2:REM CLOSE DIRECT ACCESS CHANNEL
60 CLOSE15:REM CLOSE COMMAND CHANNEL
```

Again, this example writes the new sector to the disk without regard to what existed in the sector previously. If you wish to change a byte without changing the rest of the contents in a sector, read the sector into the buffer first, change the byte, and then re-write it to the disk.

Block Commands

The BLOCK commands allow you to read, write, allocate, and de-allocate the sectors on the disk. When using these commands, you should keep in mind which type of partition you are working with, since each partition has different legal track and sector values, with the directory, BAM, and header information stored in different areas. Many of the functions performed with the BLOCK commands may also be performed with the USER commands, and in fact, the USER commands are often preferred since they do not exhibit some of the side effects of the BLOCK commands.

Allocating Blocks

The BLOCK-ALLOCATE command is used to allocate a block directly. When saving or writing files, this function is handled automatically by the DOS. BLOCK-ALLOCATE is normally used after information has been written to disk by a BLOCK-WRITE command. It can also be used to determine the next higher unallocated block available, if you try to allocate a block which has already been allocated. Upon reading the error channel, you will find that the track and sector variables of the error message will contain the track and sector of the next available block. The syntax for the BLOCK ALLOCATE command is:

```
PRINT#lf,"B-A:";n;t;s
```

where:

	lf	=	the logical file number used for the command channel
	n	=	the partition (0) in which the block to be allocated exists
	t	=	the track on which the block to be allocated exists
	s	=	the sector of the block to be allocated

Important Note: The bugs that existed with some Commodore drives with the BLOCK-ALLOCATE command do not exist with the CMD HD version of this command. There were two different bugs associated with this command. One of these caused the entire track (instead of just one sector) to be allocated on 1541 and 1571 disk drive units. The other bug caused the BAM on 1581 disk drives to be improperly allocated if another block command (such as B-W) was issued after a block allocate. Closing the file or issuing another type of disk command usually got around this problem, but these steps are unnecessary when using the HD.

Freeing Blocks

The BLOCK-FREE command allows you to free a block which is currently allocated. When scratching files, this function is handled automatically by the DOS. BLOCK-FREE is normally used to free a block in which information had been previously placed by a BLOCK-WRITE command, but which is no longer needed. The syntax for the BLOCK-FREE command is:

```
PRINT#lf,"B-F:";n;t;s
```

where:
- lf = the logical file number used for the command channel
- n = the partition (0) in which the block to be allocated exists
- t = the track on which the block to be allocated exists
- s = the sector of the block to be allocated

The Buffer Pointer

The BUFFER-POINTER command allows you to point to a specific byte within a sector which is to be read from or written to. This action occurs within the disk buffer which holds the data for the particular. The syntax for this command is:

```
PRINT#lf,"B-P";ch;pt
```

where:
- lf = the logical file number used for the command channel
- ch = the channel number used for the direct access file
- pt = the byte number within the block you wish to be used as the first byte of the next GET#, INPUT#, or PRINT# statement

The syntax given above assumes that the command and direct access channel have already been opened. The channel number is the same as the secondary address you used when opening the direct access channel. When using the BUFFER-POINTER command, remember that bytes are numbered from 0 to 255. If you need to access the first byte, it is byte 0.

Reading Blocks

The BLOCK-READ command, due to a quirk in the way it operates, is rarely used Instead, most applications use the 'U1' command, which is very similar to the BLOCK-READ command. The syntax is:

```
PRINT#lf,"B-R";ch;n;t;s
```

where:
- lf = the logical file number used for the command channel
- ch = the channel number used for the direct access file
- n = partition number (always 0)
- t = the track from which you wish to read a block
- s = the sector which you wish to read from

The channel number specified above is the same as the secondary address used to open the direct access file.

The partition number should always be 0 (zero). This is because direct access commands will always access the partition that was the current partition at the time the direct access channel was opened.

The problem with the BLOCK-READ command is that it will not read an entire block as data. Instead, it uses the first byte in the block as a counter for how many bytes are to be read. This allows you to read the full block only if the first byte of the block contains a value of 255. Since most blocks contain a link to the next sector in the first byte, any blocks intended to be read in this fashion must be written with this same consideration in mind.

Writing Blocks

The BLOCK-WRITE command, due to a quirk in the way it operates, is rarely used. Instead, most applications use the 'U2' command, which is nearly identical to the BLOCK-WRITE command. The syntax for the BLOCK-WRITE command is:

```
PRINT#lf,"B-W";ch;n;t;s
```

where:
- lf = the logical file number used for the command channel
- ch = the channel number used for the direct access file
- n = partition number (always 0)
- t = the track to which you wish to write a block
- s = the sector which you wish to write to

The channel number specified above is the same as the secondary address used to open the direct access file.

The partition number should always be 0 (zero). This is because direct access commands will always access the partition that was the current partition at the time the direct access channel was opened.

The problem with the BLOCK-WRITE command is that it will not write an entire block as data. Instead, it uses the first byte in the block as a counter for how many bytes are to be written. This will allow you to write (and later read) the full block only if the first byte of the block contains a value of 255.

Block Execute

Another rarely used command, BLOCK-EXECUTE, instructs the drive to load a specific block into drive memory, and begin executing the block as a machine language program. To do this, the DOS performs a JSR to the start of the buffer being used by the direct access channel. Unless the machine language has been written to be relocatable, this is a case where you should specify a buffer number when opening the direct access channel. The machine language routine should also end with an RTS if you wish to return control back to DOS. The syntax for this command is:

```
PRINT#lf,"B-E";ch;n;t;s
```

where:
- lf = the logical file number used for the command channel
- ch = the channel number used for the direct access file
- n = partition number (always 0)
- t = the track on which the executable block is stored
- s = the sector location of the executable block

The following example program assumes that you wish to execute a block stored at track 2, sector 0 in buffer #2 ($0500):

```
10 OPEN15,8,15
20 OPEN2,8,2,"#2"
30 PRINT#15,"B-E";2;0;2;0
40 CLOSE2:CLOSE15
```

Memory Commands

The memory commands allow you to read or write drive memory and execute machine language programs contained within drive memory. The memory commands are useful for many applications, but should be handled carefully to avoid corrupting data or important locations used by the DOS.

Reading from Drive Memory

The MEMORY-READ command allows you to read from memory locations within the HD. This command is similar to the BASIC PEEK command, with the difference being that MEMORY-READ returns drive memory instead of computer memory. Reading drive memory is useful in many applications such as determining the type of drive being used as a particular device number. The syntax for MEMORY-READ is:

```
PRINT#lf,"M-R"CHR$(ml)CHR$(mh)CHR$(nb)
```

where:
- lf = the logical file number used for the command channel
- ml = the low byte of the starting memory address
- mh = the high byte of the starting memory address
- nb = the number of bytes to be read

The value for the number of bytes to be read can be from 0 to 255. Since it is improbable that you would want to read zero bytes from drive memory, and it is also likely that you may wish to read 256 bytes on many occasions, the value of zero has been altered to mean 256.

After a MEMORY-READ command is sent, the specified bytes are returned over the error channel. Thus, you may use the GET# command to read these bytes one at a time.

To determine the low and high bytes of a decimal address, you can use the following formula:

```
HB=INT(AD/256):LB=AD AND 255
```

where:
- AD = the decimal address you wish to begin reading from
- HB = the high byte of the starting memory address
- LB = the low byte of the starting memory address

Here is an example program which determines if the drive you are reading from is an HD:

```
10 OPEN15,12,15
20 PRINT#15,"M-R"CHRS(160)CHR$(254)CHR$(6)
30 FORI=1T06:GET#15, B$:A$=A$+B$:NEXT
70 CLOSE15
80 IFA$="CMD HD"THENPRINT"HD PRESENT":END
90 PRINT"NOT A CMD HD":END
```

Note: The MEMORY-READ command will not read past a page boundary.

Writing to Drive Memory

The MEMORY-WRITE command allows you to write to memory locations within the HD. Using this command is similar to using the POKE command in BASIC, the difference being that MEMORY-WRITE writes to drive memory instead of computer memory. MEMORY-WRITE is useful for placing machine language routines into the drive, or for writing data directly to one of the buffers. The syntax for MEMORY-WRITE is:

```
PRINT#lf,"M-W"CHR$(ml)CHR$(mh)CHR$(nb)CHR$(d)...
```

where:
- lf = the logical file number used for the command channel
- ml = the low byte of the starting memory address
- mh = the high byte of the starting memory address
- nb = the number of bytes to be written
- d = value of the data byte to be written (if more than one byte is to be written use additional CHR$ statements or a string variable)

The value for the number of bytes to be written can range from 1 to 248. When sending a MEMORY-WRITE command, the bytes to be written are placed at the end of the command. This may be done in the form of CHR$ statements, or if more room is needed, as a string variable.

To determine the low and high bytes of a decimal address, you can use the formula given in the MEMORY-READ command above.

Here is an example program which illustrates the use of the MEMORY-WRITE command:

```
10 FORI-1TO248
20 A$=A$+CHR$(I)
30 NEXT
40 OPEN15,12,15
50 PRINT#15,"M-W"CHR$(0)CHR$(8)CHR$(248)A$
60 CLOSE15
70 END
```

Memory Execute

The MEMORY-EXECUTE command allows you to instruct the drive to begin execution of a machine language program located in drive memory. When this command is issued, a JSR is performed by the drive control loop. To return control of the drive to DOS, the machine language program which is being executed should end with an RTS instruction. The syntax for this command is:

```
PRINT#lf,"M-E"CHR$(ml)CHR$(mh)
```

where:	lf	=	the logical file number used for the command channel
	ml	=	the low byte of the memory address where execution should begin
	mh	=	the high byte of the memory address where execution should begin

This example executes a machine language program located at $0800 in drive memory:

```
PRINT#15,"M-E"CHR$(0)CHR$(8)
```


User Commands

The USER commands allow you to perform jumps to memory locations within the HD, execute burst utility commands, and provide useful replacements for BLOCK-READ and BLOCK-WRITE. The following text provides an overview of the burst utility commands. Additional information is given under 'Burst Commands' later in this section. All USER commands are sent via the DOS command channel (secondary address of 15).

U0 Utility Commands

The UO command has a number of uses. Without any parameters, it resets the user vectors. When combined with specific parameters, UO provides a way to send burst and utility commands. Here is the syntax:

```
PRINT#lf,"UO[+|-|>[bu]]"[+CHR$(d1)[+CHR$(d2)]]
```

where:	lf	=	the logical file number used for the command channel
	bu	=	burst utility command characters) (if needed)
	d1	=	burst utility data byte (if needed)
	d2	=	second burst utility data byte (if needed)

Since most of these commands are covered in greater detail in the paragraph titled 'CHGUTL Utility' in the area describing burst commands, we will present only the general syntax of each command here. Assume that each the command begins with a 'PRINT#*lf*,' and that the *lf* should be replaced with an already-opened logical file number for the command channel.

`"U0"`	Reset user vectors to default
`"U0+"`	*Adjust serial speed for VIC-20
`"U0-"`	*Adjust serial speed for C64/128
`"U0>"+CHR$(d)`	Change device number (d = dev.#)
`"U0>B0"`	Disable fast serial
`"U0>B1"`	Enable fast serial
`"U0>R"+CHR$(r)`	Set DOS retries (r = # of retries)
`"U0>S"+CHR$(i)`	Adjust sector interleave for 1571 burst commands (i = interleave)
`"U0>T"`	*Test ROM checksum
`"U0>V0"`	*Verify off
`"U0>V1"`	*Verify on
`"U0>M0"`	*Set 1571 single-sided mode
`"U0>M1"`	*Set 1571 double-sided mode
`"U0>M-R"+CHR$(sp)+CHR$(np)`	Burst memory read, (sp= starting page, np = number of pages)
`"U0>MW"+CHR$(sp)+CHR$(np)`	Burst memory write. (sp=starting page, np = number of pages)

*Denotes commands which are accepted but do not perform any function.

Reading Blocks with U1

The U1 command is used for reading specific blocks within a partition. U1 should be considered a direct access command, as is the BLOCK-READ command which U1 is normally used in place of. The U1 command will read an entire block into the buffer which is being used by the direct access channel. The syntax for the U1 command is:

```
PRINT#lf,"U1";ch;n;t;s
```

where: lf = the logical file number used for the command channel
ch = the channel number used for the direct access file
n = partition number (always 0)
t = the track from which you wish to read a block
s = the sector which you wish to read from

The channel number is the same as the secondary address used when opening the direct access file.

The partition number should always be 0 (zero). This is because direct access commands will always access the partition that was the current partition at the lime the direct access channel was opened.

After using the U1 command to read a block, you may retrieve the data it stores in the buffer with the GET# command. You may position the pointer with the BUFFER-POINTER command to retrieve specific bytes before using the GET# command. You may also substitute the characters UA for U1. An example of this command can be found under the heading 'Reading and Writing Data with Direct Access' earlier in this section.

Writing Blocks with U2

The U2 command is used for writing specific blocks to a partition. U2 is considered to be a direct access command, as is the BLOCK-WRITE command which U2 is normally used in place of. The U2 command will write an entire block from the buffer which is being used by the direct access channel. The syntax for the U2 command is:

```
PRINT#lf,"U2";ch;n;t;s
```

where: lf = the logical file number used for the command channel
ch = the channel number used for the direct access file
n = partition number (always 0)
t = the track to which you wish to write a block
s = the sector which you wish to write to

The channel number is the same as the secondary address used when opening the direct access file.

The partition number should always be 0 (zero). This is because direct access commands will always access the partition that was the current partition at the time the direct access channel was opened.

Before using the U2 command, you should place the data you wish to write into the direct access channel's buffer. This may be accomplished by using the PRINT# command. You may place data in specific bytes by using the BUFFER-POINTER command before sending data with PRINT#. You may also substitute the characters UB for U2. An example of this command can be found under the heading 'Reading and Writing Data with Direct Access' earlier in this section.

User Jump Commands

The majority of the remaining USER commands are less commonly used by BASIC programmers, but may be useful to machine language programmers. These commands perform jumps to certain locations in drive memory. Six of these jumps are targeted at locations in the third buffer. These jumps are performed to every third location, allowing easy setup of a user jump table. Each command also has an alias. The following table shows the User Jump commands, their alias, and the memory location which they jump to:

COMMAND	ALIAS	TARGET ADDRESS
U3	UC	$0500
U4	UD	$0503
U5	UE	$0506
U6	UF	$0509
U7	UG	$050C
U8	UH	$050F

The remaining two USER commands are used to reset the drive to various degrees. Here are their definitions:

COMMAND	ALIAS	DEFINITION
U9	UI	Warm reset (minimal effect on drive variables)
U	UJ	Cold reset (does not change current partition)

The syntax for the preceding USER commands is:

```
PRINT#lf,"Ux"
```

where: lf = the logical file number used for the command channel
x = the character of the desired USER command

Burst Commands

The HD uses two different burst command sets depending on which type of partition is being accessed. The HD Burst Command Instruction Set (HDBCIS) allows users to perform their own handshaking routines. These commands have been implemented mostly for the sake of compatibility, since there is little speed advantage in using these instructions on the HD.

1541 / 1571 / HD Native Mode Burst Commands

These commands have been setup to emulate 1571 BCIS as closely as possible. Commands or command switches which are intended for MFM formatting or parameters are not emulated in the HD.

Read

BYTE	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
00	0	1	0	1	0	1	0	1
01	0	0	1	1	0	0	0	0
02	T	E	B	X	0	0	0	N
03	DESTINATION TRACK							
04	DESTINATION SECTOR							
05	NUMBER OF SECTORS							
06	NEXT TRACK (optional)							

RANGE: See Appendix C figures C3,C4, and C14.

SWITCHES:
T - Transfer Data (1 = no transfer)
E - Ignore Error (1 = ignore)
B - Buffer Transfer Only (1 =buffer transfer only)
X - Don't Care
N - Partition Number (only 0 is supported)

PROTOCOL: Burst Handshake

CONVENTIONS: Before READING from or WRITING to a partition, it must be logged in using either the INQUIRE DISK or QUERY DISK FORMAT command. This must be done once each time you change partitions.

OUTPUT: One burst status byte, followed by burst data, is sent for each sector transferred. An error will prevent data from being sent unless the IGNORE ERROR bit in byte 02 is set to a value of 1.

Write

BYTE	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
00	0	1	0	1	0	1	0	1
01	0	0	1	1	0	0	0	0
02	T	E	B	X	0	0	1	N
03	DESTINATION TRACK							
04	DESTINATION SECTOR							
05	NUMBER OF SECTORS							
06	NEXT TRACK (optional)							

RANGE: See Appendix C figures C3,C4, and C14.

SWITCHES:
T - Transfer Data (1 = no transfer)
E - Ignore Error (1 = ignore)
B - Buffer Transfer Only (1 =buffer transfer only)
X - Don't Care
N - Partition Number (only 0 is supported)

PROTOCOL: Set fast serial output, send data, set fast serial input, pull clock low, wait for status, release clock. Repeat from beginning for multiple sector output.

CONVENTIONS: Before READING from or WRITING to a partition, it must be logged in using either the INQUIRE DISK or QUERY DISK FORMAT command. This must be done once each time you change partitions.

INPUT: Host must transfer burst data.

OUTPUT: One burst status byte following each WRITE operation.

Inquire Disk

BYTE	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
00	0	1	0	1	0	1	0	1
01	0	0	1	1	0	0	0	0
02	X	X	X	X	0	1	0	N

SWITCHES:
X - Don’t Care
N - Partition Number (only 0 is supported)

PROTOCOL: Burst Handshake

OUTPUT: One burst status byte following each INQUIRE DISK operation

Format MFM

The FORMAT MFM burst instruction is not supported in the HD. Attempts to send this command will result in a format error.

Format GCR (without directory)

BYTE	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
00	0	1	0	1	0	1	0	1
01	0	0	1	1	0	0	0	0
02	T	E	B	X	0	0	0	N
03	DESTINATION TRACK							
04	DESTINATION SECTOR							
05	NUMBER OF SECTORS							

SWITCHES:
X - Don't Care
N - Partition Number (only 0 is supported)
M - Mode (only 0 is supported)

PROTOCOL: Conventional

CONVENTIONS: This command must be followed by the INQUIRE DISK or QUERY DISK FORMAT command to log in the partition.

OUTPUT: None

Note: The ID information supplied is not actually written to the disk, since the HD does not store the ID on every track as is done on floppy diskettes.

Sector Interleave

BYTE	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
00	0	1	0	1	0	1	0	1
01	0	0	1	1	0	0	0	0
02	W	X	X	0	1	0	0	N
03	DESTINATION TRACK							

SWITCHES:
W - Write/Read (0 = write / 1 = read current)
X - Don't Care
N - Partition Number (only 0 is supported)

PROTOCOL: Burst handshake (W = 1)

CONVENTIONS: This is a soft interleave used for multiple sector burst READ and WRITE.

OUTPUT: None (W = 0). Current interleave burst byte (W = 1)

Query Disk Format

BYTE	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
00	0	1	0	1	0	1	0	1
01	0	0	1	1	0	0	0	0
02	F	X	X	X	1	0	1	N
03	OFFSET (optional - F = 1)							

SWITCHES:
F - Force Flag (F = 1 steps the head with the offset specified in byte 03)
X - Don't Care
N - Partition Number (only 0 is supported)

PROTOCOL: Burst handshake.

CONVENTIONS: Logs in partition.

OUTPUT: Burst status byte.

Inquire Status

BYTE	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
00	0	1	0	1	0	1	0	1
01	0	0	1	1	0	0	0	0
02	W	C	X	0	1	0	1	N
03	OFFSET (optional - F = 1)							

SWITCHES:
W - Write switch (W = 0 - write / W = 1 - return current status only)
C - Change (C = 1 and W = 0 - log in partition / C = 1 and W = 1 - returns whether disk was logged)
X - Don't Care
N - Partition Number (only 0 is supported)

PROTOCOL: Burst handshake (W = 1).

CONVENTIONS: This a method of reading or writing the current status.

OUTPUT: None (W = 0). Burst status byte (W = 1).

CHGUTL Utility

BYTE	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
00	0	1	0	1	0	1	0	1
01	0	0	1	1	0	0	0	0
02	X	X	X	1	1	1	1	0
03 - ??	UTILITY COMMAND AND PARAMETERS							

SWITCHES: X - Don't Care

COMMANDS:

CHR$ (n) - Change Device Number
n = new device number

Bn - Fast Serial Mode
n = 0 - fast serial disabled
n = 1 - fast serial enabled

Hn - Head Select (returns error)
n = 0 - side 0
n = 1 - side 1

Mn - Mode Select (ignored)
n = 0 - single sided
n = 1 - double sided

MR+CHR$ (a)+CHR$ (p) - Memory Read*
a = high byte of starting address
p = number of pages to read

MW+CHR$ (a)+CHR$ (p) - Memory Write*
a = high byte of starting address
p = number of pages to write

R+CHR$ (n) - DOS Retries
n = retries before error is assumed

S+CHR$ (n) - DOS Sector Interleave
n = sector interleave offset

T - Test ROM Checksum (ignored)

Vn - Verify (ignored)
n = 0 - verify on
n = 1 - verify off

*Note: Burst MR and MW use standard burst protocol (without status byte).

Fastload Utility

BYTE	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
00	0	1	0	1	0	1	0	1
01	0	0	1	1	0	0	0	0
02	P	X	X	1	1	1	1	1
03 - ??	OFFSET (optional - F = 1)							

SWITCHES: P - Non-program File Flag (P = 0 - program file / P = 1 - non-program file)
X - Don't Care

PROTOCOL: Burst handshake.

OUTPUT: Burst status byte preceding each sector transferred. In a program file, the load address should be handled in the normal manner..

STATUS:
0000000X = OK
00000010 = FILE NOT FOUND
00011111 = EOI (last sector)

Any other status value should indicate a file read error. The byte after the EOI indicates the number of data bytes remaining in the file

1581 / 1581 CP/M Burst Commands

These commands have been set up to emulate the 1581 BCIS as closely as possible. Commands or command switches which are intended for MFM formatting or parameters are not emulated in the HD.

Read

BYTE	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
00	0	1	0	1	0	1	0	1
01	0	0	1	1	0	0	0	0
02	L	E	X	S	0	0	0	N
03	DESTINATION TRACK							
04	DESTINATION SECTOR							
05	NUMBER OF SECTORS							
06	NEXT TRACK (optional)							

RANGE: Logical Format: Tracks: 01 - 80 (single sided)
Sectors: 00 - 39 (256 bytes each)
Physical Format: Tracks: 00 - 79 (double sided)
Sectors: 01 - 10 (512 bytes each)

SWITCHES:
L - Logical Flag (0 = physical / 1 = logical)
E - Ignore Error (1 = ignore)
X - Don't Care
S - Side (0 or 1 if using physical format)
N - Partition Number (only 0 is supported)

PROTOCOL: Burst Handshake

CONVENTIONS: Before READING from or WRITING to a partition, it must be logged in using either the INQUIRE DISK or QUERY DISK FORMAT command. This must be done once each time you change partitions.

OUTPUT: One burst status byte, followed by burst data, is sent for each sector transferred. An error will prevent data from being sent unless the IGNORE ERROR bit in byte 02 is set to a value of 1.

Write

BYTE	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
00	0	1	0	1	0	1	0	1
01	0	0	1	1	0	0	0	0
02	L	E	X	S	0	0	1	N
03	DESTINATION TRACK							
04	DESTINATION SECTOR							
05	NUMBER OF SECTORS							
06	NEXT TRACK (optional)							

RANGE: Logical Format: Tracks: 01 - 80 (single sided)
Sectors: 00 - 39 (256 bytes each)
Physical Format: Tracks: 00 - 79 (double sided)
Sectors: 01 - 10 (512 bytes each)

SWITCHES:
L - Logical Flag (0 = physical / 1 = logical)
E - Ignore Error (1 = ignore)
X - Don't Care
S - Side (0 or 1 if using physical format)
N - Partition Number (only 0 is supported)

PROTOCOL: Output data, set fast serial in, pull clock low, wait for status byte, release clock. Repeat for multiple sectors.

CONVENTIONS: Before READING from or WRITING to a partition, it must be logged in using either the INQUIRE DISK or QUERY DISK FORMAT command. This must be done once each time you change partitions.

INPUT: Host must transfer burst data.

OUTPUT: One burst status following each WRITE operation.

Inquire Disk

BYTE	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
00	0	1	0	1	0	1	0	1
01	0	0	1	1	0	0	0	0
02	X	X	X	X	0	1	0	N

SWITCHES:
X - Don't Care
N - Partition Number (only 0 is supported)

PROTOCOL: Burst handshake.

OUTPUT: One burst status byte following each INQUIRE DISK.

Format

BYTE	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
00	0	1	0	1	0	1	0	1
01	0	0	1	1	0	0	0	0
02	M	X	X	X	0	1	1	N
03	SECTOR SIZE (optional)							
04	LAST TRACK NUMBER (optional)							
05	NUMBER OF SECTORS (optional)							
06	STARTING TRACK (optional)							
07	FILL BYTE (optional)							
08	STARTING SECTOR (optional)							

SWITCHES:
- M - Mode 0 = logical format with directory and BAM / 1 = physical format using optional bytes 03 - 08)
- X - Don't Care
- N - Partition Number (only 0 is supported)

OPTIONS: Bytes 03 - 08 are optional and are only accepted when physical formatting is specified (M = 1). If you do not specify values for these bytes, system default values will be used. The following list shows the legal range and default values for these bytes:

Byte 03 - must be 2 (Default = 2)
Byte 04 - 00 through 79 (Default = 79)
Byte 05 - must be 10 (Default = 10)
Byte 06 - 0 through 79 (Default = 0)
Byte 07 - $00 through $FF (Default = $E5)
Byte 08 - must be 1 (Default = 1)

PROTOCOL: Conventional.

CONVENTIONS: This command must be followed by an INQUIRE DISK or QUERY DISK FORMAT command to log in the partition.

OUTPUT: None.

Note: The directory header written by this command will contain the string 'CMD HD' for the disk name and '89' for the ID.

Query Disk Format

BYTE	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
00	0	1	0	1	0	1	0	1
01	0	0	1	1	0	0	0	0
02	F	X	T	S	1	0	1	N
03	OFFSET (optional - F = 1)							

SWITCHES:
- P - Force (F = 1 steps head with offset in byte 03)
- X - Don't Care
- T - Sector Table (T = 1 sends sector table)
- S - Side
- N - Partition Number (only 0 is supported)

PROTOCOL: Burst handshake.

CONVENTIONS: Determines the partition format.

OUTPUT: Burst status byte, number of sectors (10), logical track number (0 or track number specified in byte 03), lowest logical sector number (01), highest logical sector number(10), interleave value (1), sector table (if T = 1).

Notes: The sector table is a series of 10 bytes (1,2,3,4,5,6,7,8,9,10). Status byte is from track offset 0 unless F bit is set, then status is from offset track in byte 03. No bytes follow the status byte if an error occurred.

Inquire Status

BYTE	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
00	0	1	0	1	0	1	0	1
01	0	0	1	1	0	0	0	0
02	W	C	M	0	1	1	0	N
03	NEW STATUS (W = 0)							
04	NEW OR MASK (M = 1)							
05	NEW AND MASK (M = 1)							

SWITCHES:
- W - Write (W = 0 - write new status value / W = 1 - return current status)
- C - Change (C = 1 and W = 0 - login partition / C = 1 and W = 1 - return current status)
- M - Write AND/OR masks (M = 1)
- N - Partition Number (only 0 is supported)

PROTOCOL: Burst handshake (W = 1), conventional (W = 0).

CONVENTIONS: This is a method of reading or writing the current status, and changing the status mask values.

OUTPUT: None (W = 0), burst status byte (W = 1).

CHGUTL Utility

BYTE	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
00	0	1	0	1	0	1	0	1
01	0	0	1	1	0	0	0	0
02	X	X	X	1	1	1	1	0
03 - ??	UTILITY COMMAND AND PARAMETERS							

SWITCHES: X - Don't Care

COMMANDS:

CHR$ (n) - Change Device Number
n = new device number

Bn - Fast Serial Mode
n = 0 - fast serial disabled
n = 1 - fast serial enabled

Hn - Head Select (returns error)
n = 0 - side 0
n = 1 - side 1

Mn - Mode Select (ignored)
n = 0 - single sided
n = 1 - double sided

MR+CHR$ (a)+CHR$ (p) - Memory Read*
a = high byte of starting address
p = number of pages to read

MW+CHR$ (a)+CHR$ (p) - Memory Write*
a = high byte of starting address
p = number of pages to write

R+CHR$ (n) - DOS Retries
n = retries before error is assumed

S+CHR$ (n) - DOS Sector Interleave
n = sector interleave offset

T - Test ROM Checksum (ignored)

Vn - Verify (ignored)
n = 0 - verify on
n = 1 - verify off

*Note: Burst MR and MW use standard burst protocol (without status byte).

Dump Track Cache Buffer

BYTE	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
00	0	1	0	1	0	1	0	1
01	0	0	1	1	0	0	0	0
02	F	S	X	1	1	1	0	1
03	PHYSICAL TRACK NUMBER							

SWITCHES: F - Write Always (F = 1 write even if not 'dirty')
S - Side
X - Don't Care

Note: This command is accepted, but no action takes place since the HD does not contain a track cache buffer.

Fastload Utility

BYTE	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
00	0	1	0	1	0	1	0	1
01	0	0	1	1	0	0	0	0
02	P	X	X	1	1	1	1	1
03 - ??	FILENAME							

SWITCHES: P - Non-program File Flag (P = 0 - program file / P = 1 - non-program file)
X - Don't Care

PROTOCOL: Burst handshake.

OUTPUT: Burst status byte preceding each sector transferred. In a program file, the load address should be handled in the normal manner..

STATUS:
0000000X = OK
00000010 = FILE NOT FOUND
00011111 = EOI (last sector)

Status values from 3 through 15 should be considered a file read error. The byte after the EOI indicates the number of data bytes remaining in the file.

Burst Burst Status Byte

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
M	N	S1	S0	C3	C2	C1	C0

BIT DEFINITIONS:

M - Mode: 0 Normal format
1 Foreign Disk Format (non-default physical format or default physical format without directory and BAM information)

N - Partition Number (only 0 is supported)

S - Sector Size:

S1	S0	Definition
0	0	256 bytes per sector (1541/1571/HD Native)
1	0	512 bytes per sector (1581)

C - Controller Status:

C3	C2	C1	C0	Definition
0	0	0	X	OK
0	0	1	0	Request Sense (SCSI)
0	0	1	1	Target Busy (SCSI)
0	1	0	0	Illegal Block
0	1	0	1	No Select (SCSI)
0	1	1	0	Format Error / Busy Dropped (SCSI)
0	1	1	1	Illegal Phase (SCSI)
1	0	0	0	Write Protect Error
1	0	0	1	Request Sense Failed (SCSI)
1	0	1	0	Undefined
1	0	1	1	Intermediate Status OK
1	1	0	0	Reservation Conflict (SCSI)
1	1	0	1	Hardware Error
1	1	1	0	Syntax Error / Illegal Job
1	1	1	1	No Drive Present

Special Loaders

The Commodore DOS Utility Loader and Autoboot Loader have been implemented into the DOS on the HD. This allows you to load a file into drive memory and execute it immediately by sending the name of the file preceded by an ampersand ('&') via the command channel. If a file named 'COPYRIGHT CMD 89' is present in the default partition, it is loaded and executed when the drive is reset or turned on. The syntax for the utility loader is:

```
OPENlf,dv,15:PRINT#lf,"&[n][path]:filename":CLOSElf
```

where: lf = the logical file number for the command channel
dv = the device number of the HD
n = the partition you wish to change to (1-254)
path = the Native Mode partition subdirectory path to the file
filename = the name of the file which is to be loaded and executed in drive memory.

Note: Programs written to execute in the memory of other drives will have to be modified to work with the HD, since the memory layout of the HD is much different than that of any other drive.

Job Queue Instructions

The job queue buffers are special locations used to pass commands and parameters to the drive controller for the purpose of performing specific disk access functions. The job queue can be accessed directly by the programmer if desired. Different areas are defined for job queues in the HD depending on which partition type is currently being used. This was done so that the HD could emulate the job queue locations of the 1541,1571, and 1581. In all cases, the job codes and parameters are moved to the Native Mode job queue for actual execution. When writing job queue routines for the HD, it is better to use the Native Mode job queue locations, since the routines will then operate no matter which type of partition is currently in use.

Job Queue Command Codes

Code	Name	Description
$80	READ	Reads logical block using track and sector parameters
$82	CTRL_RESET	Resets the disk controller
$84	MOTOR_ON	Not implemented (returns OK status)
$86	MOTOR_OFF	Not implemented (returns OK status)
$88	MOTORI_ON	Not implemented (returns OK status)
$8A	MOTORI_OFF	Not implemented (returns OK status)
$8C	SEEK_PHYS_OFF	Not implemented (returns OK status)
$8E	FRMT_TRK	Not implemented (returns OK status)

$90	WRITE	Writes logical block using track and sector parameters
$92	DSK_IN_DRV	Not implemented (returns OK status)
$94	ACITV_ON	Turns ACTIVITY LED on
$96	ACTIV_OFF	Turns ACTIVITY LED off
$98	ERR_ON	Turns ERROR LED on
$9A	ERR_OFF	Turns ERROR LED off
$9C	SID_SEL	Not implemented (returns OK status)
$9E	CACHE_MOV	Not implemented (returns OK status)
$A0	VERIFY	Not implemented (returns OK status)
$A2	WRT_CACHE	Not implemented (returns OK status)
$A4	PHYS_BUF0	Reads 'physical' sector to $0300
$A6	BUF0_PHYS	Writes $0300 to 'physical' sector
$A8	SEEK_PHYS	Not implemented (returns OK status)
$AA	READ_LOG_NQ	Reads logical block without moving it to the job queue buffer
$AC	WRT_LOG_NQ	Writes logical block without moving it from the job queue buffer
$B0	SEEK	Not implemented (returns OK status)
$B2	READ_PHYS_NQ	Reads 'physical' block without moving it to the job queue buffer
$B4	WRT_PHYS_NQ	Writes 'physical' block without moving it from the job queue buffer
$B6	WRT_PROT_ON	Tests write protect status. Returns $08 if write protect is on, $00 otherwise
$B8	SEEK_LOG(NLOG)	Not implemented (returns OK status)
$C0	BUMP	Seeks to logical track 0 within partition
$CC	RD_LOG_ADDR	Reads logical block to Target Address
$CE	WRT_LOG_ADDR	Writes logical block from Target Address
$D0	JUMP	Executes code in buffer
$E0	EXEC	Executes code in buffer
$EC	READ_MULTI	Reads multiple system blocks (Block Count) to Target Address
$EE	WRT_MULTI	Writes multiple system blocks (Block Count) from Target Address
$F0	FRMT_PART	Not implemented (returns OK status)
$F2	SCSI_RESET	Resets SCSI bus
$F4	UNIT_RDY	Not implemented (returns OK status)
$F6	REZERO_UNIT	Moves head of specified SCSI device and logical unit to track 0
$F8	START_UNIT	Unparks head of specified device & LUN
$FA	STOP_UNIT	Parks head of specified device & LUN
$FC	RD_PHYS_MULTI	Reads multiple physical blocks (Block Count) to Target Address
$FE	WRT_PHYS_MULTI	Writes multiple physical blocks (Block Count) from Target Address

Job Queue Locations

Address Range			Description
$0000	-	$0004	1541/1571 Emulation Mode Job Queue
$0006	-	$000F	1541/1571 Emulation Mode Track& Sector
$0002	-	$000A	1581 Emulation Mode Job Queue
$000B	-	$001C	1581 Emulation Mode Track & Sector
$0020	-	$003F	Native Mode Job Queue
$2800	-	$283F	Native Mode Track & Sector
$2840	-	$285F	'Side'
$2860	-	$287F	Target Address (high)
$2880	-	$289F	Target Address (low)
$28A0	-	$28BF	Block Count
$28C0	-	$28DF	SCSI Device
$28E0	-	$28FF	SCSI Logical Unit

Appendix A
Utilities

About the Utility Disks

A number of utility programs created by CMD are shipped with your HD. Most of these programs are located on the floppy disks that came with your system. Some of the programs, however, will be located in partition 1 on the HD itself. Be sure to backup the programs on the HD and the utility disks immediately. The label on the front side of each disk will identify the name(s) and contents of that disk. Here is a list of the programs supplied with the HD:

HD UTILITIES

HD-TOOLS.64	Partition aid for the C64 or C128 in 64 mode
HD-TOOLS.128	Partition aid for the C128
FCOPY	File copier
MCOPY	Whole disk/partition copier
1541SUB	Subdirectory creation utility
1581SUB	Subdirectory creation utility
SET HD CLOCK	Utility to set the time/date of internal clock
AUTO-BOOT 128	Boot-block creation utility for the C128
DISK CRACKER HD	Track and sector editor and drive monitor
FIX BLOCKS	Bad block replacement utility
PARK HD.64	Head parking utility for the C64
PARK HD.128	Head parking utility for the C128
BOOTQ	Used to load Q-Link software
XFER QDATA	Used to backup or transfer Q-Link member data
REWRITE DOS.64	Utility for C64 to place new DOS on the HD
REWRITE DOS.128	Utility for C128 to place new DOS on the HD
CREATE SYS.64	Utility for C64 to create full HD DOS system
CREATE SYS.128	Utility for C128 to create full HD DOS system
LLFORMAT.64	Hard disk low-level formatting utility for C64
LLFORMAT.128	Hard disk low-level formatting utility for C128
ADDDRIVE.64	Utility to add extra drive to HD system for C64
ADDDRIVE.128	Utility to add extra drive to HD system for C128
DRIVE INFO	Reports drive mechanism information

GEOS UTILITIES

HDTime	Autoexec file which sets GEOS clock
QuickMove	Partition utility for GEOS 64
QuickMove128	Partition utility for GEOS 128
CONFIGURE	Replaces GEOS 64 CONFIGURE
CONFIGURE_r	Replaces GEOS 64 CONFIGURE_r
128CONFIGURE	Replaces GEOS 128 CONFIGURE
128CONFIGURE_r	Replaces GEOS 128 CONFIGURE_r

CP/M UTILITIES

SPORTS.COM	Serial port command utility

Program Documentation

The various programs on these disks are in many cases documented in other sections of this manual. In cases where this is not true, the documentation is provided here. Some programs are public domain utilities that have proven to be useful with the HD. The public domain programs may not be fully documented here, but enough information has been included to allow those familiar with the particular type of utility to begin using it.

HD-TOOLS(.64/.128)

From 64 or 128 mode, load and run the appropriate version of HD-TOOLS. Before continuing, reset the system into Configuration Mode, as instructed on the screen. This is done by holding down the WRITE PROTECT switch and pressing RESET switch on the drive. After entering Configuration Mode, make sure that the WRITE PROTECT indicator is not lit, and if it is, press the WRITE PROTECT switch again to deactivate this function. Now press any key on the keyboard to continue to the main menu of the HD-TOOLS utility program.

Default Device Number

Select the CHANGE DEFAULT DEVICE NUMBER option from the menu. The program will display the status of all serial devices as well as the current HD default device number, and will then prompt you to select a new device number .Make your selection with the <+> and <–> keys and then press the <RETURN> key.

We recommend that you use device number 8 if the HD is to be the only or primary drive on your system. Use one of the other possible device numbers (9 through 29) if you have other drives connected. We recommend a default value of 10 if two other floppy drives are being used (remember • you can always use the SWAP functions to switch the HD device number with devices #8 and #9). You should not set the HD device number to a value which is used by another drive.

Default Partition Number

This option determines which partition will be the active partition after the HD is powered up or reset The process for setting the default partition is similar to the process for setting the default device number. The program will check to see if the partition you select is legal.

Select the CHANGE DEFAULT PARTITION NUMBER option from the menu. The program displays the current default partition number, and will then prompt you to select a new default. Make your selection with the <+> and <–> keys and press the <RETURN> key.

View Partition Table

This option shows the current status of all partitions. The <+> and <•> keys will step you through the pages in the display, and <RETURN> allows you to exit back to the main menu.

Create a New Partition

This option allows you to add new partitions to the system. The program will automatically default to the next available (unused) partition. This is usually the best choice as it will keep your partitions in order. However, if you wish, you may select any partition not yet in use. Use <+> and <-> to change the partition number, and <RETURN> to accept.

You must then select the partition type. Again, use <+> and <-> to view the available choices, and <RETURN> to accept.

If you have not selected an Emulation Mode partition, you will need to specify the partition size. The size is adjustable in increments of 256 blocks using <+> and <-> to select, and <RETURN> to accept.

The last step is to enter a name for the partition (maximum of 16 characters). Even though the usable characters have been limited in this program, you can later rename the partition with the HD DOS Rename Partition command if necessary. After entering the partition name you will be given the option to physically create the partition on the HD or to abort the process.

Delete an Old Partition

Upon choosing this option you will be shown the first partition on your HD. To select the partition that you wish to delete, use the <+> and <-> keys, and then press <RETURN> to accept. After the partition has been selected, you will be given the chance to abort the process before the partition is deleted.

IMPORTANT: If a bad (unreadable) block is encountered while a partition is being deleted, you will be in danger of losing some or all of the data stored on your HD. For this reason, you should always run the FIX BLOCKS utility before deleting partitions with HD-TOOLS. FIX BLOCKS will make sure that there are no unreadable blocks on your HD. When you run FIX BLOCKS, make sure you tell the program to check all your partitions. Refer to the description of the FIX BLOCKS program later in this section for more information.

Deleting a partition can take from a few seconds to over an hour depending on where the partition to be deleted is located on the system. Partitions located above (at a higher address than) the partition being deleted are moved down to fill the void left by the deleted partition. This is done in order to avoid the fragmentation of disk space. If you are going to delete a number of partitions, start with the last one created, and then continue in reverse order of creation. This will save a lot of time. If you want to create all new partitions from scratch, you may delete all of the current partitions by using the CREATE SYS program.

WARNING: Deleting partitions will destroy all data within those partitions. Do not delete partitions without first backing up any data you wish to keep.

Quit

Allows you to exit from HD-TOOLS. Remember to press the RESET switch on the HD before trying to use it again.

Do's and Don'ts

Do:

- plan ahead. Set up your partitions logically.
- backup important data before doing any system configuration. Chances for data loss are much greater when performing these functions

Don't:

- press RESET or any other switches on the HD while a file is open or there is any drive activity.
- Make a regular habit of partitioning. This is something you should do once or twice, and edit only when your needs change.

FCOPY

This program was created by CMD to fill the need for a file copier capable of copying any type of file between any two drives or between any two partitions on the HD. In addition, FCOPY supports Native Mode subdirectories, 1581 sub-partitions, and REU's running under RAMDOS. The use of this program is mostly self-explanatory, but we have included a breakdown of the functions here in order to provide more detail where necessary. This program may also be used to remove files, view directories, and send disk commands.

Set Source Device (F1)

This option selects the disk drive that you wish to copy files *from*. The device number and type of device will be shown on the display. If the drive type is not recognized by the program, question marks will be shown instead of the actual type. In the case of unrecognized third party disk drives, the program will work regardless unless the DOS in the particular drive happens to be highly incompatible with standard Commodore-style DOS commands and file handling procedures.

Set Target Device (F5)

This option selects the disk drive that you wish to copy files *to*. The device number and type of device will be shown on the display.

Set Source Partition (F3)

If the source device is a CMD HD, this option allows you to select the partition from which files are to be copied. The partition number, name and type will be displayed in the source area.

Set Target Partition (F7)

If the target device is a CMD HD, this will allow you to set the partition to which files are to be copied. The partition number, name and type will be displayed in the target area.

Set Source Path (S)

If the source device is a CMD HD and the partition type is Native or 1581 Emulation, or if the source device is an actual 1581, a path for subdirectories or sub-partitions may be entered by the user. Each subdirectory or sub-partition name must be separated by a slash (/).

Set Target Path (T)

Same function as Set Source Path, except that the path is intended for the Target device.

Source/Target Directory (A/B)

Allows you to view the directory of the source or target disk. The program will ask for a search pattern so that you may view files which match a given name and/or filetype. For directories read from a CMD HD, the pattern will also allow the selection of files by time and date.

Select Files (F)

This option reads the source directory into the selection buffer after you have entered a search pattern. File information is stored in a dynamic method, so if necessary, you will be able to view from 700 to well over a thousand files. If your directory contains more files than can be viewed, you will need to limit the selection by using a pattern which will match fewer files. Once you have entered the file selection mode, you may select or de-select files by pressing the RETURN key while the arrow is pointing to the file. An asterisk (*) indicates which files have been selected. Pressing the 'T' key allows you to toggle all selections ('T' will select all unselected files, and de-select all selected files). When you have finished, press the back-arrow key (←) to return to the main menu.

Reselect Files (R)

You may, at any time after selecting files, return to the file selection mode without reading the directory. This allows you to change your selections before or after copying files.

Send Disk Command (@)

This option allows you to send a disk command to either the source or the target drive. Important: disk commands do not follow the source or target path, but are instead sent to the current directory. Therefore, it is wise to make sure you are currently in the correct partition, sub-partition, or subdirectory before sending a command intended for a certain area. When sending commands to a CMD HD, you may include a partition number and path in the disk command.

Begin Copying (C)

This option starts the copying process. Only the files you have selected using the 'F' and/or 'R' options will be copied. When copying is complete, an option allows you to copy the same files to another disk (if your target drive is a floppy disk drive).

Begin Scratching (#)

This option starts scratching the files you selected using the 'F' and/or 'R' options. Please note: files can only be scratched from the source disk! You will be asked "ARE YOU SURE (Y/N)?" before the scratching operation begins.

Exit Program (←)

Allows you to exit from program. If you wish to use the program again, it must be re-loaded.

MCOPY

This program is of the type commonly referred to as a 'whole disk copier'. It can copy an entire disk between two floppy drives of the same type, a floppy drive and similar CMD HD partition , or two similar partition types on the same or separate CMD hard drives. The program will support all Commodore floppy drives, as well as any fully compatible third party drives.

MCOPY is also useful for backing up partitions to a floppy disk or to other partitions on the HD. It is also possible to copy Native Mode partitions to other Native Mode partitions of different sizes. This can be useful when you decide that you need a larger partition to hold data. Although it is also possible to copy from a larger partition to a smaller one, it is possible that some data will be lost in this process.

MCOPY is self-documenting. A help menu is always on screen and lists the available options.

SET HD CLOCK

This program allows you to set the Real Time Clock located in the HD. This program was written to be operated from the HD only. Attempts to LOAD and RUN this program from a floppy disk may cause an error.

BOOTQ

If you are a Q-Link user you should place this program in your default partition. When RUN, BOOTQ will SWAP the HD to device number 8, make partition 6 the current partition, and attempt to LOAD and RUN the special HD-compatible Q-Link software. This program must be operated from a C64 or a C128 or 128D in 64 mode. If you received your HD-compatible Q-Link software with an upgrade and did not place it in partition 6, you will have to modify this program, or LOAD the Q-Link software directly without the help of a boot program.

XFER QDATA

This program allows you to make a backup of your Q-Link member data. XFER QDATA also allows you to transfer this information from your regular Q-Link disk to the HD if you were already a Q-Link member before you received your HD-compatible Q-Link software. When backing up your Q-Link data, you should always copy the data to a floppy disk backup of the Q-Link HD software. You can make a Q-Link HD backup disk by using MCOPY to copy the partition containing the Q-Link HD software to a 1541 or 1571 disk drive.

AUTO-BOOT 128

This Public Domain utility has been included to allow you to create boot sectors for use in 128 mode. A boot sector may be created in any Emulation Mode or Native Mode partition. Native Mode partitions automatically protect the boot sector keeping it allocated at all times. Therefore, when using this program on a Native Mode partition, you will be told that the sector is already being used. If you continue with the process at this point, the program will go ahead and create the boot sector.

Before running this program, use the Change Partition command to make the partition that you wish to place the boot sector in the current partition. You may then specify drive 0 when asked for this information within the program.

PARK HD(.64/.128)

This utility is for those who connect additional drives that do not contain an auto-park feature to their HD system. When RUN, this program will park all drives attached to the CMD HD system. It is not necessary to use this utility on the original drive supplied by CMD, since this mechanism will park itself automatically when power is removed. If, however, you wish to manually park your drive with this program, it will allow you to do so.

DISK CRACKER HD

This utility is a modification of a program released to the public domain by its author, Mike Henry. DISK CRACKER HD is a disk editor and monitor intended for use only by those familiar with programs of this type. We will not make any attempt to document this software, since it should only be used by those familiar with the data storage methods used on Commodore compatible disk drives, and these individuals are usually well versed in the use of this type of software. One limitation that should be mentioned is that the program is only capable of recognizing device 8. In order to use this utility with the HD, you should change the HD's device number to 8 or use the SWAP 8 feature.

PARTITION INFO

This program provides information about all partitions on your CMD HD system. This program is provided to aid those who wish to share the HD between more than one type of computer. Creating partitions on the HD from another computer will require you to know where the Foreign Mode partitions are located. Note: you should always create your Foreign Mode partitions immediately after using the `CREATE SYS` program so that they will be located at the beginning of the HD's available storage area. Placing Foreign Mode partitions in between Commodore partitions could cause you to lose the data stored within them when partitions are deleted.

FIX BLOCKS

This program will scan the partitions you select for blocks which cannot be read from or written to without receiving an error. If such blocks are found, they will be reported by the program, and replaced with good blocks (hard disk mechanisms contain spare blocks which are not used until needed).

When a bad block is replaced, an attempt is made to transfer any data stored in the bad block to the replacement block. It is possible that data will be lost in one or two files when blocks are replaced, so be aware of this situation, and if possible, check the files in that partition for problems after a block replacement has been performed.

REWRITE DOS(.64/.128)

Your drive is shipped from the factory with HD DOS already installed. Normally, the only reason to reinstall HD DOS is when you receive an upgrade. REWRITE DOS will not destroy or alter any of your partitions or the data stored in them- it simply replaces the existing version of HD DOS. Follow the steps given below to rewrite the DOS on your HD:

1. Place the `HD UTILITIES` disk into your floppy disk drive. If you are using a non-JiffyDOS equipped system, this should be device number 8. If you are using JiffyDOS, make sure that the floppy disk drive is your default drive (use the '@#' or <CONTROL> + <D> commands).

2. LOAD and RUN the program called `REWRITE DOS.64` or `REWRITE DOS.128`, depending on which machine or mode you are using.

```
LOAD"REWRITE DOS.64", 8          (C64 without JiffyDOS)
RUN
RUN"REWRITE DOS.128"             (C128 without JiffyDOS)
↑"REWRITE DOS.64"                (C64 with JiffyDOS)
↑"REWRITE DOS.128"               (C128 with JiffyDOS)
```

3. Place the HD into CONFIGURATION MODE as instructed on the screen and press RETURN. The new operating system will be written to the system partition on the HD.

4. Press RESET on the HD when finished.

Note: REWRITE DOS should only be loaded and run from a floppy disk drive. Other programs on the floppy disk will be needed while the program is running, so do not remove the disk during this process. NEVER press the HD's RESET button while writing a new DOS.

CREATE SYS(.64/.128)

In some cases you may need to re-create the HD DOS system tables on your hard drive. This is only necessary if the system placed on the drive at the factory become corrupted or if you are replacing the original drive with a different mechanism. You may also use this program if you want to remove ALL partitions from the drive quickly. If you are using this program because of a corrupted system area, or are putting a new drive in your CMD HD, it is a good idea to use LLFORMAT first.

WARNING: CREATE SYS will destroy all data on the drive. Do not use this program unless it is absolutely necessary.

The following program describes the use of the CREATE SYS program:

1. Place the HD UTILITIES disk into your floppy disk drive. If you are using a non-JiffyDOS equipped system, this should be device number 8. If you are using JiffyDOS, make sure that the floppy disk drive is your default drive (use the '@#' or <CONTROL> + <D> commands).

2. LOAD and RUN the program called CREATE SYS.64 or CREATE SYS.128, depending on which machine or mode you are using.

```
LOAD"CREATE SYS.64", 8        (C64 without JiffyDOS)
RUN
RUN"CREATE SYS.128"           (C128 without JiffyDOS)
↑"CREATE SYS.64"              (C64 with JiffyDOS)
↑"CREATE SYS.128"             (C128 with JiffyDOS)
```

3. Place the HD into INSTALL MODE as instructed and then press RETURN.

Note: CREATE SYS should only be loaded and run from a floppy disk drive. Other programs stored on the floppy disk will be needed while the program is running, so do not remove the disk during this process. NEVER press the HD's RESET button while CREATE SYS is running.

LLFORMAT(.64/.128)

Your HD has been formatted prior to shipping, and already contains the current version of HD DOS. Should it ever become necessary, it is possible to reformat the entire drive using the program LLFORMAT, located on the HD UTILITIES disk which you received with your HD.

After reformatting, it is necessary to re-install HD DOS on the hard drive. This is accomplished by using the CREATE SYS program, also located on the HD UTILITIES disk.

LLFORMAT should also be used whenever you are adding a new drive to your HD system. Running CREATE SYS on drives which have been added is not required, since the system only resides on the first drive (the one inside the original HD case).

WARNING: Do not use LLFORMAT unless you are certain it is absolutely necessary, as this program will destroy all data on the HD.

ADD DRIVE(.64/.128)

This program allows you to add additional SCSI storage devices onto the CMD HD system. Only one device should be added at a time. After connecting a new device to the SCSI port located on the rear panel of the HD, LOAD and RUN this program. It will poll the SCSI bus for new devices. If one is found, the SCSI device and LUN will be reported and you will be asked if you wish to add this device to the system. A positive response causes the program to allocate the proper address space and SCSI device and logical unit numbers in the system tables.

After adding a new device, you should use LLFORMAT to perform low-level formatting of the device. You may then use HD-TOOLS to create partitions on this device.

1541SUB and 1581SUB

These utilities are nearly identical in function and are used to create Native Mode subdirectories which emulate the directories of 1541 and 1581 drives. Using these types of subdirectories may allow the use of programs which will not normally work in a Native Mode partition.

WARNING: Use these utilities on an empty Native Mode partition only. Any data stored in the partition will be lost

1541SUB creates a subdirectory which begins at the same location as the directory on a 1541 (track 18, sector 1). A header block for this directory is also created and placed where the header block on a 1541 is located (track 18, sector 0).

1581SUB creates a subdirectory which begins at the same location as the directory on a 1581 (track 40, sector 3). A header block for this directory is also created and placed where the header block on a 1581 is located (track40, sector 0). Due to the way Native Mode subdirectories work, 1581SUB must create two subdirectories to provide the proper headers and directory space. This program has been tested for use with Superbase and Superbase 128 and allows these programs to access larger storage areas.

To use either of these programs, LOAD and RUN the one which you wish to use. The program will ask for the device number of the HD as well as the partition number you wish to create the subdirectory in. Do not use either of these programs on a partition which contains useful data, as the partition will be formatted by the utility. You must also make sure that the partition you select for use has enough tracks to support the subdirectory (minimum 18 tracks for 1541SUB, 40 tracks for 1581SUB).

DRIVE INFO

This program will poll the SCSI drive mechanism located in your CMD HD system, and will report information concerning the manufacturer, model, storage capacity and serial number of the mechanism. DRIVE INFO does this by reading the data stored in the manufacturer's information area. When you run this program, it will ask for the device number of the HD. After you have input the device number, it will read and report the drive data.

Appendix B
Error Codes

The error codes used by the HD have been arranged to be compatible with the codes used on Commodore floppy disk drive units. Some errors, mostly those which are defined as read or write errors, have been eliminated on the HD since these conditions do not occur in the same manner on a hard drive. A number of new codes have been added as well, in order to indicate conditions which are unique to a hard drive. Whenever errors are encountered on your HD, these codes should help in localizing the problem.

Errors are returned over the command channel in the following format:

```
ec,estring,tv,sv
```

where:

	ec	=	a two-digit error number
	estring	=	an ASCII string describing the error
	tv	=	the track variable (the logical track where the error took place)
	sv	=	the sector variable (the logical sector where the error took place)

Note: Some errors define special meanings for the track and sector variables and are described below when necessary.

Command Channel Error Codes

0 0 **OK** (not an error)
This code is present when no other error condition exists.

0 1 **FILES SCRATCHED** (not an error)
Occurs after using the DOS or BASIC scratch commands. The number of files scratched will be indicated in the track variable, and the sector variable will contain a zero

0 2 **PARTITION SELECTED** (not an error)
Occurs after switching partitions with the HD DOS 'CP' command, and after changing 1581 sub-partitions with the V command. After using the 'CP' command, the track variable contains the partition number of the newly selected partition, while the sector variable contains zero. After the 7 command is issued, the track variable contains the number of the first track in the 1581 sub-partition, while the sector variable contains the number of the last track in the 1581 sub-partition.

26 **WRITE PROTECT ON**
Indicates that an attempt was made to write to the HD while the front panel write protect function was activated

30 **SYNTAX ERROR** (general)
Indicates that the HD DOS command interpreter was unable to identify the last command sent via the command channel. This is usually caused by incorrect characters being present in the disk command.

31 **SYNTAX ERROR** (unrecognized command)
Usually indicates that the first character of the last command string was not recognized as part of a legal DOS command.

32 **SYNTAX ERROR** (command string too long)
Occurs when a disk command string contains over 254 characters. Due to the large size of the command channel input buffer in the HD, this error should rarely be encountered.

33 **SYNTAX ERROR** (illegal file name)
Usually indicates that an attempt was made to use wildcards or pattern matching within a file name or disk command that does not accept wildcards.

34 **SYNTAX ERROR** (missing file name)
The last disk command failed due to a missing file name or the file name should have been preceded by a colon (:).

40 **SCSI ERROR**
The last disk operation caused an error to occur in the SCSI controller. The error message will also be accompanied by some extra parameters within the error message string. These parameters are the PETASCII representations of the SCSI error information and are presented in the following format:

```
40 "SCSI ERROR d 1 cl ek ec" TV SV
```

where:
- d = SCSI device number(0-7)
- l = SCSI logical unit number (0-7)
- cl = SCSI error class (in hexadecimal)
- ck = SCSI error key (in hexadecimal)
- ec = SCSI error code(in hexadecimal)

41 **STATUS ERROR** (target busy)
The SCSI device addressed in the last operation was busy when the access was attempted.

42 **STATUS ERROR** (intermediate status ok)
This message occurs after the successful completion of each command in a series of linked SCSI commands.

43 **STATUS ERROR** (reservation conflict)
An attempt was made to access an area on a SCSI device that was reserved for use by another host controller.

44 **CONTROLLER ERROR** (no select)
The SCSI device addressed in the last operation could not be selected

45 **CONTROLLER ERROR** (busy dropped)
The SCSI BUSY signal was dropped illegally by the intended target device.

46 **CONTROLLER ERROR** (illegal phase)
The SCSI target device requested an illegal information transfer phase. This is usually an indication of a hardware failure in either the target or host controller.

47 **CONTROLLER ERROR** (request sense failed)
The SCSI controller could not determine the nature of the last error that occurred on the selected target device. This is usually a symptom of a hardware failure within the target.

48 **ILLEGAL JOB** (in job queue)
The last job code placed into the job queue was not legal.

50 **RECORD NOT PRESENT**
The last attempt to access a relative (REL) file record specified a record number which does not yet exist This condition will result even when attempting to create or expand a relative file, and under those conditions should be ignored.

51 **OVERFLOW IN RECORD**
The last attempted write to a relative record contained more data than could be stored in the record. When this occurs, as much data as can fit is stored into the record.

52 **FILE TOO LARGE**
The last attempt to access a relative record would have exceeded the amount of storage space remaining in the partition specified. Under this condition, no additional records were added to the file, and if this was an attempt to create a new relative file, the tile was not created

60 **WRITE FILE OPEN**
The last attempt to open a file was made to a file which was already open for writing.

6 1 **FILE NOT OPEN**
Indicates that the last attempt to access data was made to a file which was not properly opened Under some circumstances, this error is not generated and the attempt to access the file is simply ignored.

6 2 **FILE NOT FOUND**
During the last attempt to open a file, the DOS could not find the file specified by the path and filename given. This error will also occur if the filetype of the file does not match the allowed filetypes.

6 3 **FILE EXISTS**
While attempting to open a new file, another file with the same name was found in the specified partition or sub-directory.

6 4 **FILE TYPE MISMATCH**
The last disk operation specified a file which did not match the filetypes allowed for that operation.

6 5 **NO BLOCK**
Indicates that an attempt was made to allocate a block which was already allocated using the BLOCK-ALLOCATE (B-A) command. The track and sector variables will contain the next available block when this condition occurs. If the track variable contains a zero, there are no blocks available.

6 6 **ILLEGAL BLOCK**
Occurs when an attempt is made to access a track or sector which does not exist This may occur if one of the track and sector links within a file have become corrupted.

6 7 **ILLEGAL BLOCK**
Usually caused by a corrupt disk. This error shows up when the block parameters for a partition do not exist

7 0 **NO CHANNEL**
This indicates that the specified channel within a disk command is already in use, or that all buffers in the drive are currently in use. This may bean indication that too many files are currently opened.

7 1 **DIRECTORY ERROR**
Indicates that the BAM (Block Availability Map) on the diskette is invalid. To correct the problem, Validate the disk.

7 2 **PARTITION FULL**
Occurs when the targeted partition or its directory are full. When this message is sent, there are still two blocks free in the partition, allowing the current file to be closed.

7 4 **DRIVE NOT READY**
Occurs after an attempt was made to access an illegal partition or a partition that has not been properly formatted.

7 5 **FORMAT ERROR**
The last format operation failed due to an error within the targeted SCSI device.

7 6 **CONTROLLER ERROR**
Indicates that a non-recoverable hardware error has occurred within the SCSI host controller.

7 7 **SELECTED PARTITION ILLEGAL**
Occurs when an attempt has been made to enter a nonexistent (or illegal) partition.

Appendix C
Partition and File Formats

When accessing individual tracks & sectors on the HD, it is important to remember that the track and sector layout is specific to the type of partition in which the access occurs. It is through this method that the HD is able to attain its high level of compatibility. Therefore, when accessing an Emulation Mode partition, the track and sector layout of the partition is identical to that of the drive that it emulates. The tables in this section should be used as a guide to indicate which tracks and sectors are available in each type of partition, and where the header, BAM and directory blocks are located. This appendix also provides information about the format of BAM and directory entries, as well as the format of different file types.

Common Formats Used in all Partition Types

DIRECTORY FILE TABLE FORMAT (ALL PARTITION TYPES) 1541 & 1571 PARTITIONS - TRACK 18 SECTOR 1 1581 PARTITION - TRACK 40 SECTOR 3 NATIVE PARTITION - TRACK 1 SECTOR 24	
BYTE	**DESCRIPTION**
0	Track pointer to next directory block (0 indicates last block)
1	Sector pointer to next directory block (255 indicates last block)
2 - 31	File entry 1 (see Figure C2)
32 - 33	Two zero (0) bytes (reserved)
34 - 63	File entry 2 (see Figure C2)
64 - 65	Two zero (0) bytes (reserved)
66 - 95	File entry 3 (see Figure C2)
96 - 97	Two zero (0) bytes (reserved)
98 - 127	File entry 4 (see Figure C2)
128 - 129	Two zero (0) bytes (reserved)
130 - 159	File entry 5 (see Figure C2)
160 - 161	Two zero (0) bytes (reserved)
162 - 191	File entry 6 (see Figure C2)
192 - 193	Two zero (0) bytes (reserved)
194 - 223	File entry 7 (see Figure C2)
224 - 225	Two zero (0) bytes (reserved)
226 - 255	File entry 8 (see Figure C2)

Figure C1

DIRECTORY FILE ENTRY FORMAT ALL PARTITION TYPES AND NATIVE MODE SUBDIRECTORIES		
BYTE	**VALUE**	**DESCRIPTION**
0		File type:
	0	DEL (Deleted)
	1	SEQ (Sequential)
	2	PRG (Program)
	3	USR (User)
	4	REL (Relative)
	5	CBM (1581 style sub-partition)
	6	DIR (Native Mode subdirectory)
		Note: Filetypes will be OR'ed with $80 when the file has been properly closed. Filetypes will be OR'ed with $C0 if the file is locked.
1		Track pointer to first data block (or header block if filetype is DIR)
2		Sector pointer to first data block (or header block if filetype is DIR)
3 - 18		Filename padded with shifted spaces ($A0)
19		Pointer to starting track of side sector or super side sector if filetype is REL
20		Pointer to starting sector of side sector or super side sector if filetype is REL
21		Record length if filetype is REL
22		Reserved
23	0	Year file was created (last two digits)
24		Month file was created
25		Day file was created
26		Hour file was created
27		Minute file was created
28		Number of blocks used by file (low byte)
29		Number of blocks used by file (high byte)

Figure C2

1541 and 1571 Emulation Mode Partitions

SECTORS PER TRACK (1541 EMULATION MODES)		
TRACK RANGE	**SECTORS AVAILABLE**	**TOTAL**
1 through 17	0 through 20	21
18 through 24	0 through 18	19
25 through 30	0 through 17	18
31 through 35	0 through 16	17

Figure C3

SECTORS PER TRACK (1571 EMULATION MODES)		
TRACK RANGE	**SECTORS AVAILABLE**	**TOTAL**
1 through 17	0 through 20	21
18 through 24	0 through 18	19
25 through 30	0 through 17	18
31 through 35	0 through 16	17
36 through 52	0 through 20	21
53 through 59	0 through 18	19
60 through 65	0 through 17	18
66 through 70	0 through 16	17

Figure C4

HEADER & BAM (1541 & 1571 EMULATION MODES) TRACK 18 SECTOR 0		
BYTE	**VALUE**	**DESCRIPTION**
0	18	Track pointer to first directory block
1	1	Sector pointer to first directory block
2	65	ASCII 'A' for format type
3	0	1541 Emulation Mode
	128	1571 Emulation Mode
4 - 143		BAM (Block Availability Map)
144 - 161		Disk name padded with shifted spaces
162 - 163		Disk ID
164	160	Shifted Space for separator
165	50	ASCII '2' for DOS version
166	65	ASCII 'A' for format type
167 - 170	160	Shifted Space for separator
171 - 220	0	Null bytes - reserved
221 - 255	0	1541 Emulation Mode - Null bytes - reserved 1571 Emulation Mode - Number of sectors available for tracks 36 through 70 - one byte per track (part of 1571 (side 2) BAM)

Figure C5

BAM for 1571 (side 2) EMULATION MODE TRACK 53 SECTOR 0		
BYTE	**VALUE**	**DESCRIPTION**
0 - 104		BAM for tracks 36 through 70 (3 bytes per track)
105 - 255	0	Null bytes - reserved

Figure C6

BAM ENTRY FORMAT 1541 & 1571 (side 1) EMULATION MODES Format of bytes 4-143 in Track 18 Sector 0 (Figure C5) 4 bytes per track: bytes 4-7 cover track 1, bytes 8-11 track 2, ...	
BYTE	**DESCRIPTION**
0	Number of sectors available on track
1	Block Availability for sectors 0 - 7
2	Block Availability for sectors 8 - 15
3	Block Availability for sectors 16 - 23
Notes:	The lowest bit (LSB) in each byte (bytes 1 through 3) indicates the status of the lowest sector covered by that byte. A binary value of 1 indicates that the sector is available, while a value of 0 indicates that the sector is allocated.

Figure C7

BAM ENTRY FORMAT 1571 (side 2) EMULATION MODES Format of bytes 0-143 on Track 53 Sector 0 (Figure C6) 3 bytes per track: bytes 0-2 cover track 36, bytes 3-5 track 37, ...	
BYTE	**DESCRIPTION**
0	Block Availability for sectors 0 - 7
1	Block Availability for sectors 8 - 15
2	Block Availability for sectors 16 - 23
Notes:	The lowest bit (LSB) in each byte (bytes 0 through 2) is used to indicate the status of the lowest sector covered by that byte. A binary value of 1 indicates that the sector is available, while a value of 0 indicates that the sector is allocated. The associated byte for the number of sectors available on each track is stored in bytes 221 through 225 of track 18 sector 0 (see Figure C5).

Figure C8

1581 Emulation Mode Partitions

SECTORS PER TRACK		
TRACK RANGE	**SECTORS AVAILABLE**	**TOTAL**
1 through 80	0 through 39	40

Figure C9

DIRECTORY HEADER TRACK 40 SECTOR 0		
BYTE	**VALUE**	**DESCRIPTION**
0	40	Track pointer of first directory block
1	3	Sector pointer of first directory block
2	68	ASCII 'D' for format type
3	0	Reserved
4 - 21		Disk name padded with shifted spaces
22 - 23		Disk ID
24	160	Shifted Space for separator
25	51	ASCII '3' for DOS version
26	68	ASCII 'D' for format type
27 - 28	160	Shifted spaces for separators
29 - 255	0	Null bytes - reserved

Figure C10

BAM BLOCK1 TRACK 40 SECTOR 1		
BYTE	**VALUE**	**DESCRIPTION**
0	40	Track pointer to next BAM block
1	2	Sector pointer to next BAM block
2	68	ASCII 'D' for DOS version
3	187	Complement of version number
4 - 5		Disk ID
6	192	Not used in HD - set to 1581 default value
7	0	Flag for Auto Loader file
8 - 15	0	Reserved
16 - 255		BAM for tracks 1 through 40 (6 bytes per track)

Figure C11

BAM 2 TRACK 40 SECTOR 2		
BYTE	**VALUE**	**DESCRIPTION**
0	0	Indicates last sector for BAM
1	255	Indicates all bytes in sector used
2	68	ASCII 'D' for DOS version (copy)
3	187	Complement of version number (copy)
4 - 5		Disk ID (copy)
6	192	Not used in HD - set to 1581 default value
7	0	Flag for Auto Loader file (copy)
8 - 15	0	Reserved
16 - 255		BAM for tracks 41 through 80 (6 bytes per track)

Figure C12

BAM ENTRY FORMAT Format of bytes 16 - 255 in Track 40 Sectors 1 and 2 (Figures C11 & C12)	
BYTE	**DESCRIPTION**
0	Number of sectors available on track
1	Block Availability for sectors 0 - 7
2	Block Availability for sectors 8 - 15
3	Block Availability for sectors 16 - 23
4	Block Availability for sectors 24 - 31
5	Block Availability for sectors 32 - 39
Notes:	The lowest bit (LSB) in each byte (bytes 1 through 5) indicates the status of the lowest sector covered by that byte. A binary value of 1 indicates that the sector is available, while a value of 0 indicates that the sector is allocated.

Figure C13

Native Mode Partitions

SECTORS PER TRACK		
TRACK RANGE	**SECTORS AVAILABLE**	**TOTAL**
1 through 255	0 through 255	256

Figure C14

ROOT DIRECTORY AND SUBDIRECTORY HEADER TRACK 1 SECTOR FOR ROOT DIRECTORY VARIES FOR SUBDIRECTORIES		
BYTE	**VALUE**	**DESCRIPTION**
0		Track pointer of first directory block
1		Sector pointer of first directory block
2	72	ASCII 'H' for format type
3	0	Reserved
4 - 21		Disk name padded with shifted spaces
22 - 23		Disk ID
24	160	Shifted spaces for separator
25	49	ASCII '1' for DOS version
26	72	ASCII 'H' for format type
27 - 28	160	Shifted Space for separators
29 - 31	0	Reserved
32	1	Pointer to ROOT header track
33	1	Pointer to ROOT header sector
34		Track pointer to DIR PARENT header
35		Sector pointer to DIR PARENT header
36		Track pointer to DIR entry in PARENT directory
37		Sector pointer to DIR entry in PARENT directory
38		Index pointer to DIR entry in PARENT directory
39 - 255	0	Null bytes - reserved

Figure C15

NATIVE MODE BAM (1st BAM block) TRACK 1 SECTOR 2		
BYTE	VALUE	DESCRIPTION
0	0	Reserved
1	0	Reserved
2	72	ASCII 'H' for format type
3	183	Complement of format type
4 - 5		Disk ID
6	192	Not used in HD - set at 1581 default value
7	0	Flag for Auto Loader file
8		Track number of last available track in partition
9 - 31	0	Reserved
32 - 255	0	BAM for tracks 1 through 7 (32 bytes per track)

Figure C16

NATIVE MODE BAM (blocks 2-32) TRACK 1 SECTORS 3 - 28		
TRACK	SECTOR	DESCRIPTION
1	3	BAM for tracks 8 through 15 (32 bytes per track)
1	4	BAM for tracks 16 through 23 (32 bytes per track)
1	5	BAM for tracks 24 through 31 (32 bytes per track)
1	6	BAM for tracks 32 through 39 (32 bytes per track)
1	7	BAM for tracks 40 through 47 (32 bytes per track)
1	8	BAM for tracks 48 through 55 (32 bytes per track)
1	9	BAM for tracks 56 through 63 (32 bytes per track)
1	10	BAM for tracks 64 through 71 (32 bytes per track)
1	11	BAM for tracks 72 through 79 (32 bytes per track)
1	12	BAM for tracks 80 through 87 (32 bytes per track)
1	13	BAM for tracks 88 through 95 (32 bytes per track)
1	14	BAM for tracks 96 through 103 (32 bytes per track)
1	15	BAM for tracks 104 through 111 (32 bytes per track)
1	16	BAM for tracks 112 through 119 (32 bytes per track)
1	17	BAM for tracks 120 through 127 (32 bytes per track)
1	18	BAM for tracks 128 through 135 (32 bytes per track)
1	19	BAM for tracks 136 through 143 (32 bytes per track)
1	20	BAM for tracks 144 through 151 (32 bytes per track)
1	21	BAM for tracks 152 through 159 (32 bytes per track)
1	22	BAM for tracks 160 through 167 (32 bytes per track)
1	23	BAM for tracks 168 through 175 (32 bytes per track)
1	24	BAM for tracks 176 through 183 (32 bytes per track)
1	25	BAM for tracks 184 through 191 (32 bytes per track)
1	26	BAM for tracks 192 through 199 (32 bytes per track)
1	27	BAM for tracks 200 through 207 (32 bytes per track)
1	28	BAM for tracks 208 through 215 (32 bytes per track)
1	29	BAM for tracks 216 through 223 (32 bytes per track)
1	30	BAM for tracks 224 through 231 (32 bytes per track)
1	31	BAM for tracks 232 through 239 (32 bytes per track)
1	32	BAM for tracks 240 through 247 (32 bytes per track)
1	33	BAM for tracks 248 through 255 (32 bytes per track)

Figure C17

NATIVE MODE BAM ENTRY FORMAT Format of bytes 32 - 255 in Track 1 Sector 2 and bytes 0 - 255 in Track 1 Sectors 3 - 33 (Figures C11 & C12)	
BYTE	**DESCRIPTION**
0	Block Availability for sectors 0 - 7
1	Block Availability for sectors 8 - 15
2	Block Availability for sectors 16 - 23
3	Block Availability for sectors 24 - 31
4	Block Availability for sectors 32 - 39
5	Block Availability for sectors 40 - 47
6	Block Availability for sectors 48 - 55
7	Block Availability for sectors 56 - 63
8	Block Availability for sectors 64 - 71
9	Block Availability for sectors 72 - 79
10	Block Availability for sectors 80 - 87
11	Block Availability for sectors 88 - 95
12	Block Availability for sectors 96 - 103
13	Block Availability for sectors 104 - 111
14	Block Availability for sectors 112 - 119
15	Block Availability for sectors 120 - 127
16	Block Availability for sectors 128 - 135
17	Block Availability for sectors 136 - 143
18	Block Availability for sectors 144 - 151
19	Block Availability for sectors 152 - 159
20	Block Availability for sectors 160 - 167
21	Block Availability for sectors 168 - 175
22	Block Availability for sectors 176 - 183
23	Block Availability for sectors 184 - 191
24	Block Availability for sectors 192 - 199
25	Block Availability for sectors 200 - 207
26	Block Availability for sectors 208 - 215
27	Block Availability for sectors 216 - 223
28	Block Availability for sectors 224 - 231
29	Block Availability for sectors 232 - 239
30	Block Availability for sectors 240 - 247
31	Block Availability for sectors 248 - 255
Notes:	the lowest bit (LSB) in each byte (bytes 0 through 31) indicates the status of the highest sector covered by that byte. A binary value of 1 indicates that the sector is available, while a value of 0 indicates that the sector is allocated.

Figure C18

File Formats

PROGRAM FILE FORMAT	
BYTE	**DESCRIPTION**
0	Pointer to track of next file block (contains a zero if current block is last block in file).
1	Pointer to sector of next file block (contains pointer to last byte used if current block is last block in file).
2 - 255	Program data (bytes 2 and 3 contain load address of program in low byte-high byte format if current block is first block in file).

Figure C19

SEQUENTIAL FILE FORMAT	
BYTE	**DESCRIPTION**
0	Pointer to track of next file block (contains a zero if current block is last block in file).
1	Pointer to sector of next file block (contains pointer to last byte used if current block is last block in file).
2 - 255	Data bytes.

Figure C20

RELATIVE FILE DATA BLOCK FORMAT	
BYTE	**DESCRIPTION**
0	Pointer to track of next data file block (contains a zero if current block is last data block in file).
1	Pointer to sector of next data file block (contains pointer to last byte used if current block is last data block in file).
2 - 255	Data bytes .Empty records will begin with a $FF in the first byte of the record, the remaining bytes will contain $00 bytes. Partially filled records will also contain $00 bytes in the unused portion of the record.

Figure C21

RELATIVE FILE SUPER SIDE SECTOR BLOCK FORMAT	
BYTE	**DESCRIPTION**
0	Pointer to track of first side sector in first group (group 0).
1	Pointer to sector of first side sector in first group (group 0).
2	Super side sector identification byte ($FE)
3 - 255	Track and sector pointers to first side sector of 126 groups (groups 0 through 125, two bytes per pointer). Unused group pointers contain $00 bytes.

Figure C22

RELATIVE FILE SIDE SECTOR BLOCK FORMAT	
BYTE	DESCRIPTION
0	Pointer to track of next side sector in this group (contains a zero if current block is last side sector block in use).
1	Pointer to sector of next side sector in this group (contains pointer to last byte used if current block is last side sector block in use).
2	Side sector number (0 - 5)
3	Record length of associated relative file
4	Pointer to track of first side sector in this group (number 0).
5	Pointer to sector of first side sector in this group (number 0).
6 - 7	Pointer to track and sector of second side sector in this group (number 1).
8 - 9	Pointer to track and sector of third side sector in this group (number 2).
10 - 11	Pointer to track and sector of fourth side sector in this group (number 3).
12 - 13	Pointer to track and sector of fifth side sector in this group (number 4).
14 - 15	Pointer to track and sector of sixth side sector in this group (number 5).
16 - 255	Track and sector pointers to 120 data blocks (two bytes per pointer). Unused data block pointers contain $00 bytes.

Figure C23

Appendix D
HD Memory Map

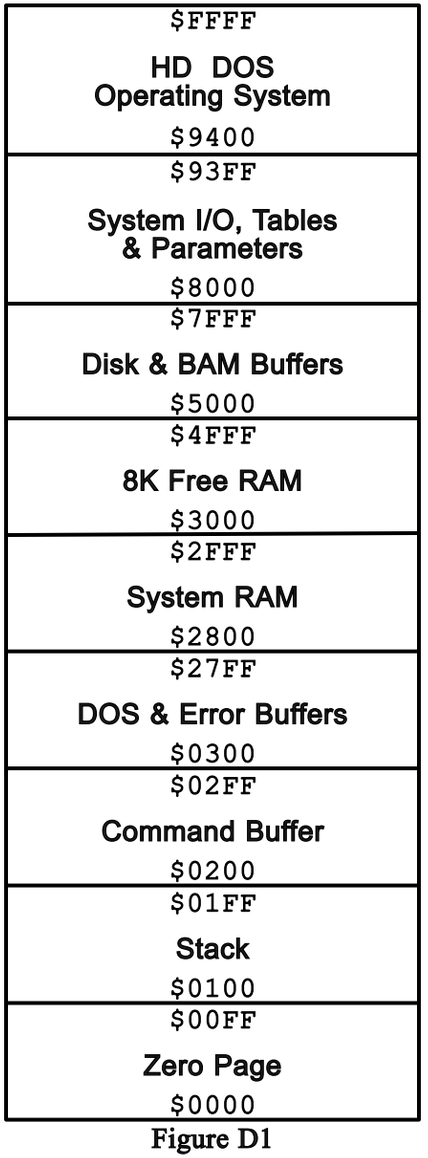


Figure D1

CMD HD Extended Memory Map

Address Range	Description of Contents
$0000 - $0004	1541/1571 Emulation Mode Job Queue
$0006 - $000F	1541/1571 Job Queue Track & Sector Variables
$0002 - $000A	1581 Emulation Mode Job Queue
$000B - $001C	1581 Job Queue Track & Sector Variables
$0020 - $003F	Native Mode Job Queue
$0040 - $00FF	Zero Page Variables
$0100 - $01FF	Processor Stack
$0200 - $02FF	Input Command Buffer (254 bytes)
$0300 - $22FF	DOS Buffers (32 buffers - 256 bytes each)
$2300 - $27FF	Error Buffers
$2800 - $283F	Native Mode Job Queue Track & Sector Variables
$2840 - $285F	Side Variable
$2860 - $287F	Target Address (high) Variable
$2880 - $289F	Target Address (low) Variable
$28A0 - $28BF	Block Count Variable
$28C0 - $28DF	SCSI Device for Job
$28E0 - $28FF	SCSI Logical Unit for Job
$2900 - $291F	SCSI Block Address (high) Variable (Error Class)
$2920 - $293F	SCSI Block Address (middle) Variable (Error Key)
$2940 - $295F	SCSI Block Address (low)Variable (Error Code)
$2960 - $297F	Last Job in Job Queue
$2980 - $29FF	Reserved
$2A00 - $2AFF	SCSI Controller Variables
$2B00 - $2FFF	DOS Variables
$3000 - $4FFF	Free RAM
$5000 - $5FFF	Reserved For Disk Buffers
$6000 - $7FFF	BAM Buffers
$8000 - $81FF	Serial I/O (6522)
$8200 - $82FF	Partition Type Table
$8300 - $83FF	Partition Size (high) Table
$8400 - $85FF	SCSI I/O (6522)
$8600 - $86FF	Partition Size (low) Table
$8700 - $87FF	Partition Starting Address (high) Table
$8800 - $89FF	Parallel I/O (8255)
$8A00 - $8AFF	Partition Starting Address (middle) Table
$8B00 - $8BFF	Partition Starting Address (low) Table
$8C00 - $8DFF	Real Time Clock I/O
$8F00 - $8FFF	LED I/O
$9000 - $93FF	DOS Tables
$9400 - $FFFF	Operating System

Note: All partition-related tables contain one byte per partition starting with partition 0 (System partition) and ending with partition 254.

Appendix E
SCSI Port

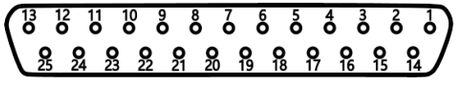


Figure E1

SCSI Port Pin Definitions			
Pin Number	Function	Pin Number	Function
1	$\overline{REQ}$	14	GND
2	$\overline{MSG}$	15	$\overline{C}$/D
3	$\overline{I}$/O	16	GND
4	$\overline{RESET}$	17	$\overline{ATN}$
5	$\overline{ACK}$	18	GND
6	$\overline{BSY}$	19	$\overline{SEL}$
7	GND	20	$\overline{DBP}$
8	$\overline{DB0}$	21	$\overline{DB1}$
9	GND	22	$\overline{DB2}$
10	$\overline{DB3}$	23	$\overline{DB4}$
11	$\overline{DB5}$	24	GND
12	$\overline{DB6}$	25	N/C
13	$\overline{DB7}$		

Figure E2

Appendix F
Serial and Auxiliary Ports

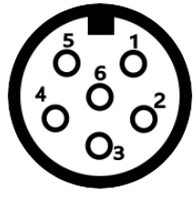


Figure F1

SERIAL PORT PIN DEFINITIONS		
PIN NO.	PIN NAME	DESCRIPTION
1	$\overline{\text{SRQ}}$	SERVICE REQUEST
2	GND	GROUND
3	$\overline{\text{ATN}}$	ATTENTION
4	$\overline{\text{CLK}}$	CLOCK
5	$\overline{\text{DATA}}$	SERIAL DATA
6	$\overline{\text{RESET}}$	SERIAL BUS RESET

Figure F2

AUXILIARY PORT PIN DEFINITIONS		
PIN NO.	PIN NAME	DESCRIPTION
1		NO CONNECTION
2	GND	GROUND
3	$\overline{\text{AUX ATN}}$	AUXILIARY ATTENTION
4	$\overline{\text{AUX CLK}}$	AUXILIARY CLOCK
5	$\overline{\text{AUX DATA}}$	AUXILIARY DATA
6	$\overline{\text{AUX RESET}}$	AUXILIARY BUS RESET

Figure F3

Appendix G
Parallel Port

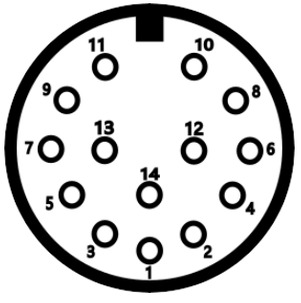


Figure G1

PIN NO.	PIN NAME	DESCRIPTION
1	$\overline{\text{PCLK}}$	PARALLEL CLOCK
2	GND	GROUND
3	$\overline{\text{PEXT}}$	RESERVED
4	PD1	PARALLEL DATA BIT 1
5	PD4	PARALLEL DATA BIT 4
6	GND	GROUND
7	$\overline{\text{PRDY}}$	PARALLEL READY
8	PD0	PARALLEL DATA BIT 0
9	PD7	PARALLEL DATA BIT 7
10	$\overline{\text{PATN}}$	PARALLEL ATTENTION
11	PD5	PARALLEL DATA BIT 5
12	PD3	PARALLEL DATA BIT 3
13	PD6	PARALLEL DATA BIT 6
14	PD2	PARALLEL DATA BIT 2

Figure G2

Appendix H
Power Connector

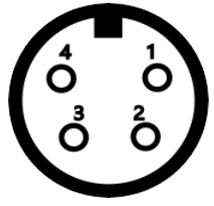


Figure H1

PIN NO.	DESCRIPTION
1	+5 VDC
2	GROUND
3	GROUND
4	+12 VDC

Figure H2

Appendix I
Drive Specifications

Storage

System Capacity (maximum)	4,244,766,720 bytes
(typical)	>20,000,000 bytes (HD-20)
	>40,000,000 bytes (HD-40)
	>100,000,000 bytes (HD-100)
	>200,000,000 bytes (HD-200)
Maximum number of partitions:	254 (user partitions)
Partition sizes (logical sectors):	256 to 65280blocks (Native)
	684 (1541- 683 usable)
	1366(1571)
	3200(1581)
Maximum Sequential file size:	16,564,864 bytes (Native)
Maximum Relative file size:	~16,000,000 bytes (Native)
Records per Relative file:	65,535
Files per partition:	>59,000 (Native)
	144 (1541 & 1571)
	296(1581)
Logical sector size	256 bytes
Physical sector size	512 bytes

Integrated Circuit Chips

Microprocessor:	6502A(2MHz clock)
Input/Output:	6522A(2),8255
ROM:	16K (27128 or half of 27256)
RAM:	TMS 4464-12 (2,64K bytes total)
Time Clock:	RTC 72421A
Custom Chips:	16L8PAL (3)
Glue:	16 MOS & TTL

Physical Dimensions

Height:	2.75in/70mm
Width:	5.00in/127mm
Depth:	10.00in/254mm
Weight:	8 lbs. (approx.)

Electrical Requirements (North America)

Voltage:	117 VAC (nominal)
Frequency:	60 Hz (nominal)
Power Consumption:	30 Watts (typical)

Appendix J
Command Summary

Partition Commands

BOOT (pg. 9-16)

```
BOOT[[Dn]{ON|,}[Udv]]
```

CHANGE PARTITION (pg. 9-10)

```
OPENlf,dv,15:PRINT#lf,"CPn":CLOSElf
OPENlf,dv,15:PRINT#lf,"C❐"+CHR$(n):CLOSElf
```

CHANGE SUB-PARTITION (1581) (pg. 9-11)

```
OPENlf,dv,15:PRINT#lf,"/[n]:[partname]":CLOSElf
```

CREATE SUB-PARTITION (1581) (pg. 9-9)

```
OPENlf,dv,15:PRINT#lf,"/[n]:partname,"CHR$(st)
CHR$(ss)CHR$(sl)CHR$(sh)",C":CLOSE15
```

FORMAT (pg. 9-11)

```
OPENlf,dv,15:PRINT#lf,"N[n]:partname[,id]"
:CLOSElf
HEADER"partname"[,Iid][,Dn][{ON|,}Udv]
```

GET PARTITION INFORMATION (pg. 9-15)

```
OPENlf,dv,15:PRINT#lf,"G-P"[+CHR$(n)]:CLOSElf
```

INITIALIZE (pg. 9-13)

```
OPENlf,dv,15:PRINT#lf,"I[n][:]":CLOSElf
```

PARTITION DIRECTORY (pg. 9-14)

```
LOAD"$=P[:*][=tp]",dv
```

RENAME PARTITION (pg. 9-14)

```
OPENlf,dv,15:PRINT#lf,"R-P:newname=oldname":CLOSElf
```

RENAME HEADER (pg. 9-15)

```
OPENlf,dv,15:PRINTlf,"R-H[n][path]:newname":CLOSElf
```

VALIDATE (pg. 9-13)

```
OPENlf,dv,15:PRINT#lf,"V[n][:]":CLOSElf
COLLECT[,Dn][{ON|,}Udv]
```

Subdirectory Commands

CHANGE DIRECTORY (pg. 9-18)

```
OPENlf,dv,15:PRINT#lf,"CD[n]{[<-]|[[path][:]
subname]}":CLOSElf
```

MAKE DIRECTORY (pg. 9-17)

```
OPENlf,dv,15:PRINT#lf,"MD[n][path]:name":CLOSElf
```

REMOVE DIRECTORY (pg. 9-19)

```
OPENlf,dv,15:PRINT#lf,"RD[n]:subname":CLOSElf
```

File Commands

COPY (pg. 9-28)

```
OPENlf,dv,15:PRINT#lf,"C[n][path]:newfile=[[n]
[path]:]filename[,[n:]filename...]":CLOSElf
COPY[Dn,]"filename"TO[Dn]"newfile"[,Udv]
CONCAT[Dn,]"filename"TO[Dn]"newfile"[,Udv]
```

LOAD (pg. 9-23)

```
LOAD"[[n][path]:]filename",dv[,sa]
BLOAD"filename"[,Dn][{ON|,}Udv][,Bb][,Pa]
DLOAD"filename"[,Dn][{ON|,}Udv]
```

LOAD (Directory File) (pg. 9-20)

```
LOAD"$=[T][n][path][:pattern[={tp|option}
[,option]]],dv
```


LOCK (UNLOCK) (pg. 9-30)

```
OPENlf,dv,15:PRINT#lf,"L[n][path]:name":CLOSElf
```

RENAME (Files and Subdirectories) (pg. 9-26)

```
OPENlf,dv,15:PRINT#lf,"R[n][path):newname=[[n]
[path]:]filename":CLOSElf
RENAME[Dn,]"filename"TO[Dn]"newfile"[,Vdv]
```

SAVE (pg. 9-24)

```
SAVE "[[@][n][path]:]filename",dv
BSAVE"[@]filename"[,Dn][{ON|,)Udv][,Bb],Pa TO Pe
DSAVE"[@]filename"[,Dn][{On|,}Udv]
```

SCRATCH (pg. 9-27)

```
OPENlf,dv,15:PRINT#lf,"S[n][path]:filename[,[n]
[path]:]filename...]":CLOSElf
SCRATCH"filename"[,Dn][{On|,}Udv]
```

VERIFY (pg. 9-25)

```
VERIFY"[[n][path]:]filename",dv[,sa]
DVERIFY"filename"[,Dn][{ON|,}Udv]
```

Relative File Commands

OPEN (REL file) (pg. 9-31)

```
OPENlf,dv,sa,"[[n][path]:]filename[{"|,L"
+CHR$(rl)}]
DOPEN#lf,"filename"[,Lrl][,Dn][,Udv]
```

POSITION (pg. 9-32)

```
PRINT#lf,"P"+CHR$(ch)+CHR$(lr)+CHR$(lh)
[+CHR$(of)]
RECORD#lf,rn[,of]
```

Special HD Commands

S-C (Send SCSI Command) (pg. 9-39)

```
OPENlf,dv,15:PRINT#lf,"S-C"CHR$(de)CHR$(bl)
CHR$(bh)CHR$(cb)...
```

SWAP (pg. 9-34)

```
OPENlf,dv,15:PRINT#lf,"S-x":
FORt=1TO500:NEXT:CLOSElf
```

Real Time Clock Commands

T-RA (Read Time in ASCII) (pg. 9-36)

```
OPENlf,dv,15:PRINT#lf,"T-RA"
```

T-WA (Write Time in ASCII) (pg. 9-37)

```
OPENlf,dv,15
PRINT#lf,"T-WAdw. mo/da/yr hr:mi:se xM"
CLOSElf
```

T-RD (pg. 9-37)

```
OPENlf,dv,15:PRINT#lf,"T-RD"
```

T-WD (pg. 9-38)

```
OPENlf,dv,15
PRINT#lf,"T-WD"+CHR$(byte0)+CHR$(byte1)+
CHR$(byte2)+CHR$(byte3)+CHR$(byte4)+CHR$(byte5)+
CHR$(byte6)+CHR$(byte7):CLOSElf
```

T-RB (pg. 9-38)

```
OPENlf,dv,15:PRINT#lf,"T-RB"
```

T-WB (pg.9-38)

```
"T-WB"+BCD time
```

WRITE PROTECT (pg. 9-35)

```
OPENlf,dv,15:PRINT#lf,"W-x":CLOSElf
```


Direct Access Commands

OPEN (Direct Access Channel) (pg.9-40)

```
OPENlf,dv,sa,"#[bu]
```

BLOCK-ALLOCATE (pg. 9-43)

```
PRINT#lf,"B-A:";n;t;s
```

BLOCK-EXECUTE (pg. 9-46)

```
PRINT#lf,"B-E:";ch;n;t;s
```

BLOCK-FREE pg. 9-44)

```
PRINT#lf,"B-F:";n;t;s
```

BLOCK-READ (pg. 9-44)

```
PRINT#lf,"B-R";ch;n;t;s
```

BLOCK-WRITE (pg. 9-45)

```
PRINT#lf,"B-W";ch;n;t;s
```

BUFFER-POINTER (pg. 9-44)

```
PRINT#lf,"B-P";ch;pt
```

MEMORY-READ (pg. 9-46)

```
PRINT#lf,"M-R"CHR$(ml)CHR$(mh)CHR$(nb)
```

MEMORY-WRITE (pg. 9-47)

```
PRINT#lf,"M-W"CHR$(ml)CHR$(mh)CHR$(nb)CHR$(d)...
```

MEMORY-EXECUTE (pg. 9-48)

```
PRINT#lf,"M-E"CHR$(ml)CHR$(mh)
```

UO (Utility Commands) (pg. 9-49)

```
PRINT#lf,"UO[+|-|>[bu]]"[+CHR$(d1)[+CHR$(d2)]]
```

"U0"	Reset user vectors to default
"U0+"	*Adjust serial speed for VIC-20
"U0-"	*Adjust serial speed for C64/128

"U0>"+CHR$(d)	Change device number
"U0>B0"	Disable fast serial
"U0>B1"	Enable fast serial
"U0>R"+CHRS(r)	Set DOS retries
"U0>S"+CHR$(i)	Adjust sector interleave for 1571 burst commands
"U0>T"	*Test ROM checksum
"U0>V0"	*Verify off
"U0>V1"	*Verify on
"U0>M0"	*Set 1571 single-sided mode
"U0>M1"	*Set 1571 double-sided mode
"U0>MR"+CHR$(sp)+CHR$(np)	Burst memory read
"U0>MW"+CHR$(sp)+CHRS(np)	Burst memory write

*Denotes commands which are accepted but do not perform any function.

U1 (BLOCK-READ) (pg.9-50)

```
PRINT#lf,"Ul";ch;n;t;s
```

U2 (BLOCK-WRITE) (pg.9-50)

```
PRINT#lf,"U2";ch;n;t;s
```

USER JUMP COMMANDS (pg. 9-51)

```
PRINT#lf,"Ux"
```


LIMITED WARRANTY

Creative Micro Designs, Inc., 50 Industrial Dr., P.O. Box 646, East Longmeadow, Massachusetts warrants to the original retail purchaser of the CMD HD hard drive that it is free of defects in material and workmanship for a period of six months from date of purchase from an authorized CMD dealer or six months from the date of delivery if purchased direct from CMD.

IMPLIED WARRANTIES, OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR OTHERWISE, ARE LIMITED IN DURATION TO THE DURATION OF THE EXPRESS WARRANTY SET FORTH ABOVE. IN NO EVENT SHALL CMD BE LIABLE FOR ANY LOSS, INCONVENIENCE, OR DAMAGE WHETHER DIRECT. INCIDENTAL. CONSEQUENTIAL OR OTHERWISE RESULTING FROM BREACH OF ANY EXPRESS OR IMPLIED WARRANTY. OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR OTHERWISE, WITH RESPECT TO THE EQUIPMENT, EXCEPT AS SET FORTH HEREIN.

SOME STATES DO NOT ALLOW THE LIMITATIONS ON THE LIFE OF AN IMPLIED WARRANTY. SOME STATES MAY ALSO DISALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATIONS OR EXCLUSIONS MAY NOT PERTAIN TO YOU.

DESCRIPTION OF WARRANTY RIGHTS

From the date of purchase or date of delivery , in the case of a direct sale through CMD, through the warranty period, CMD will, at its discretion, repair or replace any part deemed to be defective at no charge for parts/labor to the original retail customer. During the applicable warranty period wherein CMD will repair or replace defective parts without charge for labor, all warranty inspections and repairs must be performed at a CMD authorized service agency or by CMD itself.

CONDITIONS TO WARRANTY SERVICE

For this warranty to become effective the following requirements must be met:

1. Any postage, insurance and shipping charges of warranted items to a CMD authorized service agency or CMD itself must be prepaid by the original retail purchaser and these costs are not included under this warranty. Return shipping during the warranty period will be paid by CMD to addresses in the continental U.S. All other addresses will be charged for shipping, insurance and any other charges related to the return shipping of the item.
2. The dealer's original bill of sale or a charge or credit or delivery receipt must be retained by the original retail purchaser as proof of purchase date of the warranted item and must be presented to the CMD authorized service agency or CMD itself when warranty claims are advanced.
3. The warranty registration card must be filled out and returned to CMD within 30 days of purchase. If CMD does not receive, in good condition, the warranty registration card within the 30 day period, all warranty services are forfeited by the original retail purchaser.

4. Any CMD HD hard drive being returned for warranty repairs must be in its original shipping container or one of equivalent structure. An equivalent shipping container must be able to maintain a 30 G internal shock rating.

EXCLUSIONS FROM THE WARRANTY

This warranty does not cover the specific items/or conditions described below:

1. Equipment which has been damaged due to:
   - Accident, misuse, abuse, fire, flood, or "Acts of God" or other contingencies beyond the control of CMD.
   - Use of incorrect line voltages.
   - Improper or insufficient ventilation.
   - Failure to follow CMD's operating instructions.
   - Improper or unauthorized repair's.
   - Any unauthorized modification to the drive or device to which it is attached.
   - Improper return packaging or damages caused by failure to insure
2. Damage to warranted items sustained in shipment to the original retail purchaser.
3. Power transformer voltage or Power Supply conversion to foreign or domestic voltage or current frequency.
4. Any damage resulting from the infection of the unit by a computer virus..
5. Routine adjustments.
6. Any unit which has had its serial number altered, defaced or removed.
7. Any unit which has had its rear panel sticker altered, removed or defaced.
8. Damage resulting from the commercial use of this unit.

CMD will not be responsible for labor charges of unauthorized service agencies. CMD will not be responsible for labor charges from CMD authorized service agencies or CMD itself except during the warranty period applicable thereto. CMD will not be responsible for the loss or damage to equipment while in the possession of a CMD authorized service agency. CMD reserves the right to make changes in its design and improvements upon its product without assuming the obligation to install such changes on any of its products previously manufactured However, all customers purchasing the HD with version 1 of the operating system will be entitled to a free upgrade to the next version of the operating system. With this offer CMD reserves the right to charge a shipping and handling fee if it so chooses.

This warranty gives you specific legal rights and you may also have other rights which vary state to state.

RETURN POLICY

This unit may be returned to Creative Micro Designs, Inc. within 30 days of purchase for a refund of the purchase price less a 10% restocking fee. Shipping charges and taxes are not refundable.

Goods being returned must be returned in original condition in the original shipping container, freight prepaid, and must also include all accessories and be accompanied by a letter stating the reason for return. This letter should contain a return authorization number obtained from Creative Micro Designs, Inc. The return authorization number should also be clearly visible in large characters on the shipping carton.

Return authorization numbers may be obtained by contacting Creative Micro Designs, Inc. by telephone during normal business hours.

Index

A

access lime 1-2, 1-3, A-l, A-ll
accessories 1-2
add-on drive 1-3, 1-4
ADD DRIVE 2-5
allocating blocks 9-43
Amiga 1-3
AUTO-BOOT 128 A-l, A-8
auto-park 2-3, A-8
autoboot loader 9-65
autobooting 9-16
auxiliary port 3-7

B

BAM 4-2,4-3,9-43, C-4-C-6, C-8-C-10
BASIC 7.0 9-2, 9-4
BBS 1-4
benefits 1-2
block commands 9-43
boot configuration 3-1
boot ROM 2-2
BOOTQ 7-2, 9-34, A-l, A-7
bug reports 1-5
burst commands 4-3, 9-52-9-64

C

C-SCAN+ 1-3
cables 1-4
  parallel 1-4
  serial 1-4, 2-1
capacity 1-3, 4-1, 4-2, 4-3, 6-1, A-12
Codehead 1-5
cold reset 9-51
combining files 9-28
commands
  BASIC
    CLOSE 9-31, 9-35
    GET# 9-6, 9-41, 9-50
    INPUT# 9-6, 9-41
    OPEN 9-31
     PRINT# 9-31, 9-33, 9-41, 9-42, 9-51

commands (cont.)
BASIC 7.0 9-17, 9-23
BLOAD 9-24
BOOT 9-16
BSAVE 9-25
COLLECT 9-13
CONCAT 9-29
COPY 9-29
DCLOSE 9-31
DLOAD 9-24
DOPEN 9-31, 9-32
DSAVE 9-25
DVERIFY 9-26
HEADER (FORMAT) 9-11, 9-12
RECORD 9-32, 9-33
RENAME 9-26
SCRATCH 9-27
burst 4-4, 9-52-9-59
command switches 9-52
status byte 9-64
direct access
BLOCK-ALLOCATE 943
BLOCK-EXECUTE 9-46
BLOCK-FREE 944
BLOCK-READ 9-41, 9-44-9-45
BLOCK-WRITE 9-41, 9-42, 9-43, 9-44, 9-45
BUFFER-POINTER 9-44
file 23
COPY 9-28, 9-29
LOAD 9-23
LOCK 9-30
RENAME 9-26
SAVE 9-24
SAVE with Replace 9-24, 9-25
SCRATCH 9-27
UNLOCK 9-30
VERIFY 9-25-9-26
JiffyDOS
Change Directory 9-19
Change Partition 9-10
1581 style 9-11
CONTROL-D 9-4
COPY 9-29
INITIALIZE 9-13
LOAD 9-4, 9-8, 9-23
directory with time and date 9-22
partition directory 9-14

commands
JiffyDOS (cont.)
LOCK/UNLOCK 9-30
Make Directory 9-18
NEW(Format) 9-12
Remove Directory 9-19
RENAME 9-26
RENAME-HEADER 9-15
RENAME-PARTITION 9-14
SAVE 9-24
SCRATCH 9-27
SWAP 9-35
VALIDATE 9-13
VERIFY 9-25
WRITE PROTECT 9-35
memory
MEMORY-EXECUTE 9-48
MEMORY-READ 9-41, 9-46-9-47
MEMORY-WRITE 9-42, 9-47-9-48
partition
Change Partition 9-10
1581 style 9-11
GET-PARTITION Info 9-15
INITIALIZE 9-13
NEW(FORMAT) 9-11-9-12
RENAME-HEADER 9-15
RENAME-PARTITION 9-14
VALIDATE 9-13
Relative file 9-31
POSITION 9-32, 9-33
special 9-34
real lime clock 9-36-9-38
SCSI Send 9-39
SWAP 9-34-9-35
Write protect 9-35
subdirectory
Change Directory 9-18, 9-19
Make Directory 9-17
Remove Directory 9-19
User 9-49-9-51, 9-56,9-62
commercial software 1-3, 4-5
types of 8-1
using 8-1
Commodore DOS commands 4-1
Commodore Information Network 1-4
communications parameters 1-4
compatibility 1-1, 1-5, 4-2, 4-5

compatibility problems 1-5
configuration 2-4
  parameters 2-4
Configuration Mode 3-6, A-2, A-10
copy-protection 3-1, 8-1, 8-3
copying files 9-28
  using JiffyDOS 9-28
COPYRIGHT CMD 89 9-65
CP/M 1-5, 4-4
  boot block 4-4
  CCP.COM 6-3
  command file 6-3
  command parameters 6-3
  commands
    Control-C 6-3
    FORMAT 6-2
    SPORT 6-3
  F0RMAT.COM 6-2
  installation 6-1
  partitions 4-2, 4-4, 6-1
    1581 CP/M 4-4
    changing 6-1, 6-3
    Emulation Mode
      1581 4-4
    logging 6-3
  sending DOS commands 6-3
  SPORT.COM 6-2, 6-3, A-2
  UTILITIES disk 6-3
CREATE SYS 2-3, 3-6, 3-7, A-l, A-4, A-9, A-10, A-ll
current partition 4-5, 8-3, 9-3, 9-8, 9-11, 9-12, 9-15, 9-16t 9-17, 9-21,
  9-25, 9-26,9-29,9-32,9-40,9-45. 9-50, 9-51, A-7, A-8

D

date 3-2, 4-3
default device number 2-4, 3-1, 3-7, 7-2, 9-34, A-2, A-3
default partition 2-4, 3-1, 3-7, 9-65, A-3, A-7
deleting files 9-27
direct access 9-6, 9-41-9-46
  reading data 9-41
  writing data 9-42
direct access channel 9-40, 9-44, 9-45, 9-46, 9-50, 9-51
direct access commands 9-8 ,9-40, 9-45, 9-50, 9-51
direct access file 9-8, 9-40, 9-45, 9-50
directories C-l, C-2, C-7
  renaming headers 9-15
  viewing 9-20
  viewing time and date 9-21

disk access 3-2, 9-6
disk commands 3-3, 4-5
DISK CRACKER HD A-l, A-8
disk drives
  1541 8-2, A-ll
  1571 4-3
  1581 4-3, 5-1, 8-2
DOS 1-2, 1-5, 3-6
DOS wedge 1-2
DRIVE INFO A-l, A-12
drive mechanism 1-2, 2-3, 2-6

E

error codes B-l
errors 9-7, 9-33, 9-40
  error channel 9-7
  reading 9-6
  SCSI 9-39, B-2
  variables
    DS 9-7
    DS$ 9-7

F

fast serial 4-3, 4-4
FCOPY 7-2, 8-1, 8-2, 8-3, 9-28, A-l, A-5
features 1-1
file formats C-ll
file type C-2
FIX BLOCKS A-l, A-4, A-9
FPPS Mode 3-3, 3-4, 7-3
  changing partitions 3-3
  indicator lamps 3-3
  setting digits 3-3
  switches
    SWAP8 3-3, 3-4
    SWAP 9 3-3, 3-4
    WRITE PROTECT 3-3, 3-4
freeing blocks 9-44
front panel
  indicators 2-2
    ACTIVITY 2-2, 3-6
    ERROR 2-2
    SWAP8 2-2
    SWAP9 2-2
    WRITE PROTECT 2-2

front panel (cont.)
  switches 3-3
    RESET 3-1, 3-6
    SWAP 3-1, 3-2
      SWAP 8 3-1, 3-6
      SWAP9 3-1, 3-6
    SWAP8 3-3
    SWAP9 3-3
    WRITE PROTECT 3-2, 3-3, 3-6
Front Panel Partition Selection Mode 3-3

G

GEOS 1-5, 2-2, 3-2
  autoexec file 5-3, A-2
  booting 5-7
  CONFIGURE 5-1, 5-2, 5-3, 5-4, 5-5, 5-8, A-2
  HDTime 5-1, 5-3, 5-4
  installing 5-2
  partitions 5-4, 5-5
    accessing 5-5, 5-6
    copying files 5-1, 5-5
    current 5-6, 5-7
    viewing 5-6
  save configuration 5-4

H

hard drives 1-2, 1-4, 1-5
  MFM 1-2
  RLL 1-2
HD DOS 4-1
  installation 2-3
HD UTILITIES 2-3, 2-4, 2-6, 3-1, 3-6, 7-2, 7-3, A-l, A-9, A-10, A-ll,
  A-1, A-2, A-4, A-11
HD-TOOLS 2-4, 3-1, 3-6, 3-7, 4-1, 4-4, 4-5, 6-1, 6-3, 9-5, 9-9, 9-10, 9-11
HDTime 5-8, A-2
hookup 2-1
housekeeping 2-6

I

IBM 1-3
IBM-compatible 1-2
information 1-4
Install Mode 3-6, A-10

J

JiffyDOS 1-2, 1-5, 9-7
JiffyDOS file-copier 9-28
job queue 9-65-9-67

L

LLFORMAT 2-3, A-l, A-10, A-ll
loading files 9-23
locking files 9-30
low-level formatting 2-3
Lt. Konnector 1-3

M

Macintosh 1-3, 2-5
manual 1-5
MCOPY 7-1, 7-2, 7-3, 8-1, 8-2, 8-3, 9-28, A-l, A-7, A-8
megabytes 1-3
MFM parameters 9-52, 9-58
motor heat 2-6
motor noise 2-6
MS-DOS 4-5
multiple-disk programs 3-3
mutiplexer 1-3

O

orders 1-4

P

parallel 2-2
  port 3-7
PARK HD 2-3, A-l, A-8
partition directory 9-14
PARTITION INFO 4-4
partitioning 2-4, A-4
  1581 style sub-partitions 4-4, 9-11
  software 4-4
partitions 2-4, 3-3, 3-4, 3-6, 3-7, 4-1, 4-4, 4-5, 7-3, 8-1, 8-2, 9-4, 9-8, 9-9,
  9-10, 9-14, 9-16, 9-31, 9-40, 9-52, 9-53, 9-58, 9-59, A-3, A-4, A-7, A-9
  1581 style
    creating 9-9
  backing up A-7
  changing 3-3, 8-2, 9-10
    FPPS Mode 3-3
  cchecking A-4, A-9
  copying files A-5

partitions (cont.)
creating 9-9, 9-11, A-3, A-9, A-11
deleting 9-10, A-4, A-9
deleting all A-4, A-10
direct access 9-40
Emulation Mode 2-4, 4-1, 4-2, 8-1, 8-2, 9-31, A-3, C-1, C-3, C-5
1541 4-2, 4-3, 7-2, 7-3, 8-1, 9-16
1571 4-2, 4-3, 4-4, 8-1, 9-16
1581 4-2, 4-4, 5-4, 7-3, 8-1, 9-9, 9-11, 9-12, 9-16
initializing 9-11, 9-13
sub-partitions 9-9
1581 CP/M 4-2, 4-4, 9-16
boot sector A-8
Foreign Mode 4-1, 4-4
creating A-9
formatting 9-11
getting partition information 9-15
guidelines 2-4
information A-9
initializing 9-13
Native Mode 2-4, 4-1, 4-2, 4-5, 8-2, 8-3, 9-2, 9-16, 9-31, A-8, C-7
backing up A-7
boot sector A-8
Print Buffer 4-1, 4-4
renaming 9-14
validating 9-13
viewing A-3
pattern matching 9-20
peripherals 1-3, 1-4
power 2-2
power supply 1-3
power switch 2-2, 2-3
powering up 2-2
powerup diagnostics 2-2

Q

Q-Link 1-4, 1-5, 7-1-7-3
QuickMove 5-1, 5-5, 5-6, 5-7, 5-8, A-2
Quantum Computer Services 1-5

R

RAM 2-2
RAM expansion 1-3
RAMDOS 1-3, A-5
RAMLink 1-3, 1-4, 3-7
reading blocks 9-44, 9-50

reading drive memory 9-46
reading time and date 9-36-9-38
real time clock 2-2, 2-5
battery 2-5
setting 2-6
Relative files 9-31-9-33
creating 9-31
opening 9-31
positioning to records 9-32
renaming files 9-26
REU 5-1, 5-5, A-5
REWRITE DOS 2-3, 3-6, A-1, A-9, A-10

S

saving files 9-24
scratching files 9-27
SCSI 1-1, 1-2, 1-3, 2-2, 2-3, 2-5
device numbers 2-5
errors 9-39, B-2
port 2-5
pin configuration 2-5
sending SCSI commands 9-39
serial 9-49
bus 1-3, 3-1, 3-2, 9-34, 9-35
device status 9-7
mode selection 9-56, 9-62
port 2-1, 2-2
SET HD CLOCK 2-6, A-1, A-7
software 1-5
special directory commands 4-3
special loaders 9-65
speed 1-2
speed enhancement 1-2
status 9-7
storage capacity 1-3
storage media 1-2
sub-partitioning
1581 style 9-11
sub-partitions (1581 style) 4-5, 9-9, 9-10, 9-11, A-5
copying files A-5
creating 9-9
deleting 9-10
formatting 9-12
moving between 9-11
subdirectories 4-1, 4-2, 4-5, 8-2, C-7
1541SUB 7-3, 8-2, A-1, A-11, A-12
1581SUB 8-2, A-1, A-11, A-12

subdirectories (cont.)
copying files A-5
creating 9-17
deleting 9-19
guidelines 9-17
moving between 9-18
nesting 4-6
paths 9-2, 9-4, 9-17, 9-18, 9-19, 9-20, 9-23
using JiffyDOS 9-4
renaming 9-26
super side sectors 9-31
support files 1-4
SWAP 3-2
SWAP function 3-1
Swapping Device Numbers 3-1
switches
front panel 3-3
system capacity 2-5
system configuration 1-5, A-4

T

technical assistance 1-4
telecommunication service 1-4
telecommunications services 1-4
time 3-2, 4-3

U

utilities 1-4, 1-5, 8-2
unlocking files 9-30
User commands 9-43
utility loader 9-65

V

verifying files 9-25

W

warm reset 9-51
write protect 3-2
writing blocks 9-45, 9-50
writing time and date 9-37, 9-38
writing to drive memory 9-47

X

XFER QDATA 7-3, A-1, A-8

Problem Reporting Log

If a problem with using your HD should occur, or if you need advice or wish to inform CMD of any findings you have discovered about the drive or its use with specific software packages, please complete the following form and mail to Creative Micro Designs, Inc.

Name: _____

Address: _____

City: _____ State: _____ Zip Code: _____

Telephone #: _____

Drive Type: _____ Serial #: _____

Detailed Description of Condition:

Instructions to Reproduce Condition:
(If possible please supply CMD with a copy of software or hardware in question)

If more room is required, please staple to additional paper